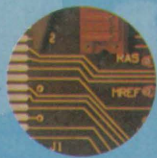
UNA PUBBLICAZIONE
DEL GRUPPO EDITORIALE JACKSON
ANNO 2 N. 5
Bit
MICROPROCESSORS-HARDWARE - SOFTWARE
HOME & PERSONAL COMPUTERS
L. 2000
Spedizione in abb. postale Gruppo IV/70
Nisi

NANOCOMPUTER® Z80

Sistema basato sulla CPU Z80 studiato dalla SGS-ATES espressamente per impieghi didattici.

- **IL PIU' POTENTE SISTEMA DIDATTICO SUL MERCATO**
  4K di RAM, 2K di ROM, interfaccia per terminale seriale e cassette magnetiche, 4 porte di I/O, tastiera a 26 tasti, display a 8 digit, accessibilità al bus completa.
- **UTILIZZABILE ANCHE PER SVILUPPO HARDWARE**
  Una scheda addizionale contenente un breadboard senza saldature e dotata di interruttori ed indicatori luminosi, permette di sviluppare circuiti di interfaccia di crescente complessità.
- **MASSIMA FLESSIBILITA' ED ESPANDIBILITA'**
  Espansione sulla scheda fino a 16K di RAM, 8K di ROM, USART, stampante parallela, espansione attraverso schede addizionali fino a 64K di RAM/ROM, interfaccia video e floppy disk.
- **NON SOLO UN MANUALE DI ISTRUZIONE**
  Tre libri in italiano, pensati come parte integrante del sistema.
- **COMPLETO SUPPORTO SOFTWARE E HARDWARE**
  Un monitor da 2K, assembler/editor/debugger, BASIC, tutto su una sola scheda. Kit di espansione, alimentatori, schede per esperimenti, schede a wire wrap, connettori, cavi ...

Bit

SOMMARIO

Vol. 5 Nº 5 - Novembre/Dicembre 1979

In copertina:
La grafica col personal computer.
Grafica di Renato NISI.

INDICE INSERZIONISTI

DIRETTORE RESPONSABILE
Giampietro Zanga

COORDINATORE TECNICO
Marcello Montedoro

CAPO REDATTORE
Dino Bortolossi

SEGRETARIA DI REDAZIONE
Cecilia De Serio

GRAFICA E IMPAGINAZIONE
Job Line srl

CONSULENZA E COORDINAMENTO
Marcello Montedoro

DIFFUSIONE E ABBONAMENTI
Gabriella Napoli, Silvia Decari
Marco Benedetti

DIREZIONE, REDAZIONE
P.le Massari, 22 - 20125 Milano
Telefoni 68.03.68 - 68.00.54

AMMINISTRAZIONE
Via Vincenzo Monti, 15 - 20123 Milano

PUBBLICITA': Concessionario per l'Italia e l'Estero Reina & C. S.n.c.
Via Ricasoli, 2 - 20121 Milano
Tel. (02) 803101 - 866192 - 8050977
Telex: 320419 BRUS I

FOTOCOMPOSIZIONE: New Comp
Via S. Michele al Carso, 5
Nova Milanese

STAMPA: Litografia del Sole srl
Buccinasco

Concessionario esclusivo per la diffusione in Italia e all'Estero:
SODIP - Via Zuretti, 25 - 20125 Milano

Spedizione in abbonamento Postale Gruppo IV/70

Prezzo della rivista L. 2.000

Numero arretrato L. 3.000
Abbonamento annuo L. 8.000
per l'Estero L. 12.000

I versamenti vanno indirizzati a: Jackson Italiana Editrice S.r.l. Via V. Monti, 15 - 20123 Milano mediante emissione di assegno bancario, cartolina vaglia o utilizzando il c/c Postale numero 11666203
Per i cambi d'indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

EDITORIALE

Gli occhi lucidi

Già, e anche il naso che cola. Sono cose che succedono a chi si commuove per qualche ragione, come ad esempio per aver scoperto che il mondo è un tantino migliore di quel che si pensava: a noi di BIT è successo.

Il fatto è che, dopo un anno di attività, nessuno dei lettori "normali" (ossia privi di diretti interessi commerciali) si era degnato di scriverci un rigo, non dico di complimenti, ma nemmeno di insulti. Neanche i lettori più giovani, quelli in cui l'entusiasmo non è stato ancora spazzato via dalle pedate nei denti che usualmente la vita somministra. Così, mogi mogi, ci eravamo ripetuti le solite cose trite e ritrite come "stiamo allevando una generazione di imbranati" oppure "viviamo in un paese in cui chi sa leggere non sa scrivere e chi sa scrivere non ne ha voglia" eccetera eccetera.

Bé, è bastato un editoriale appena un po' esplicito, quel tanto da farvi capire che ci pare giusto che chi ha un'idea la dica, e siamo stati clamorosamente smentiti da una carrettata di lettere sveglie, piacevoli, con idee anche ottime ("ma-perché-non-ci-abbiamo-pensato-noi-lo-che-vi-pago-a-fare").

Ci avete scritto in qualche centinaio, chi su lussuosa carta intestata, chi su extra strong, chi su fogli di quaderno: prevalentemente giovani o giovanissimi, quasi tutti entusiasti. E nessuno ha scritto per riempiere un pezzo di carta, per dirci bravi e basta: tutti avete portato un contributo di idee, di critiche, di suggerimenti, di opinioni, di pareri. In un attimo ci è passata l'impressione di parlare al muro che da un anno ci pesava sul groppone. Ci tocca, questo sì, rimangiarci la prosopopea da vecchi so-tutto e l'errore di avervi giudicati tonti: evidentemente non lo siete e ritrattiamo volentieri con molte scuse.

La prima necessità che ci avete fatto presente (ma che comunque sarebbe stata implicita, dato il numero delle lettere) è quella di trovare sulla rivista uno spazio apposito, che vi possa servire da punto di incontro e di confronto di idee; abbiamo pensato di espandere FEEDBACK e di affidarlo ad un tipo che ha scelto la bizzarra professione di risolutore di problemi; ha un pessimo carattere ma speriamo che vi piaccia ugualmente.

Per quel che riguarda gli altri suggerimenti, vi rimandiamo appunto a FEEDBACK, dove c'è abbastanza spazio per esaminarli tutti. Ma ciò non significa che il feedback da parte vostra debba fermarsi qui: se quel che avete da dire non sta in una lettera, se avete inventato un qualche diabolico marchingegno e volete farlo sapere al mondo, se su un argomento ne sapete abbastanza e vi pare che non sia stato trattato a sufficienza, fatene un articolo, lo pubblicheremo. E non state troppo a preoccuparvi delle virgole, a quelle, se volete, pensiamo noi.

Forse per questo tipo di collaborazione non vi pare sufficiente la nostra gratitudine e la soddisfazione di veder pubblicate le vostre fatiche; siete in questo caso dei venali materialisti, come i nostri collaboratori fissi. OK, sanguisughe: verrete ricompensati con vile, sporco e turpe denaro (a meno che non preferiate un grazioso e pulitissimo assegno), purché il materiale che ci mandate da pubblicare sia valido e sia farina del vostro sacco. Scritti e schemi vi verranno comunque restituiti, se lo volete, perché sappiamo quanta fatica costi farli.

No, avete capito male, non vi chiediamo di sostituirvi a noi; non intendiamo sederci e lasciare che siate voi a mandare avanti la rivista, vogliamo invece impegnarci ancora di più. È ormai un segreto di Pulcinella il fatto che nel 1980 usciremo con 8 fascicoli e che stiamo pedalando a tutta forza per diventare mensili. Vale a dire: su le maniche, voi e noi, e diamoci sotto! Vogliamo che BIT rimanga la rivista delle cose fattibili, degli schemi effettivi, dei programmi che si possono capire e che funzionano, dei computers che non costano milioni di dollari. Intendiamo occuparci di fatti concreti, non di sogni eterei o di Verità eterne ed inafferrabili. Siete ancora qui, sfaticati? Spicciatevi a leggere la rivista e poi mettetevi sotto a pensare a qualcosa di interessante. No, aspettate un momento: sapete che vi dico? Grazie.

Gianni Longo

SINTESI

Personal computer: un fatto di istruzione innanzitutto

Con questo numero BIT compie un anno. Al nostro attivo vantiamo cinque numeri di una rivista che, per quello che ci è dato verificare, ha avuto notevoli consensi, ed una mostra sui personal computer (BIT 79) che moltissimi hanno visitato.

Una premessa personale: quando un anno fa ho accolto l'invito dell'Editore per la direzione tecnica di BIT ho ritenuto indispensabile avviare con il lettore un discorso che, pur se a volte arduo, lo portasse a valutare nel suo insieme questo fatto nuovo che è la possibilità di avere a portata di mano uno strumento che non a caso si chiama computer (o calcolatore o elaboratore).

Il computer non è nato oggi, e la sua storia merita di essere tenuta presente giusto perchè è estremamente indicativo vedere come le sue caratteristiche, le sue possibilità sono state di *generazione* in *generazione* ridefinite. Il momento attuale, che vede l'estensione dell'elaborazione dati ad un'area estremamente composita e come interessi e come problemi da risolvere, ha tutte le caratteristiche di una ridefinizione dell'uso del calcolatore: benchè vi siano numerosi prodotti, non sono ben chiare complessivamente le strategie di ottimizzazione utenza-prodotto.

Ciò che ha reso possibile l'allargamento dell'elaborazione dati ad un'utenza che tende sempre più ad aumentare è stato da una parte la caduta dei costi di produzione della componentistica elettronica (perfezionamento dei processi di integrazione), dall'altra l'estensione di questo processo a circuiti sempre più complessi, quali le unità di calcolo di un computer (microprocessore).

Parallelamente, anche se per ragioni e con tempi diversi, si è avuta la collocazione del calcolatore in ambienti in cui l'interazione utente-macchina era una condizione essenziale.

Orbene, di questa caratteristica operativa, il personal computer se ne è impossessato in maniera completa. La visibilità del computer, una volta limitata a pochi eletti, dietro ai quali vi era poi un'altra serie di addetti ai lavori, ed infine l'utente con il suo pacco di schede perforate, è ora finalmente completa.

E' chiaro che le caratteristiche del prodotto computer sono anche notevolmente cambiate: l'hardware è molto semplificato, l'interfacciamento semplice, il software di base abbastanza elementare.

Al di là di ogni altra considerazione, tutto ciò pone le premesse perchè venga realizzata l'effettiva utilizzazione del computer in aree nuove, *personali.* Ma una condizione è indispensabile: l'utente deve impossessarsi dei termini e delle conoscenze necessarie per poter trattare con un oggetto che, per quanto semplificato, è e rimane un computer.

Personalmente ritengo che l'allargamento delle conoscenze sui calcolatori sia l'elemento più *rivoluzionario* portato dall'elaborazione personale, giusto l'importanza che in futuro l'elaborazione elettronica avrà.

Comunque anche per altre ragioni si impone l'istruzione dell'utente, e sono le caratteristiche della distribuzione del prodotto personal computer. Anche in questo c'è stato un capovolgimento; come conseguenza l'insieme di *servizi* di cui un calcolatore, grande o piccolo che sia, necessita, è stato rivoluzionato. La distribuzione del *prodotto* personal computer è incentrata, piuttosto che su filiali della casa madre o su organizzazioni che hanno a che fare tradizionalmente con i calcolatori, su *negozi* che, se hanno il merito di una distribuzione capillare, hanno il limite di servire l'utente solo per l'aspetto hardware (assistenza, accessori, etc.), o per l'aspetto software applicativo, là dove questo presenta caratteristiche di generalizzazione evidenti (giochi, applicazioni tecniche standard, etc.), essendo l'intervento specifico limitato o dalla carenza di personale specializzato o dal costo elevato dell'intervento. E' evidente quindi la differenza notevole, anche per questo aspetto, in rapporto ai prodotti classici dell'informatica; poco male: la migliore software house siamo noi stessi!

Il personal computer non è come l'Hi-Fi o la televisione a colori, che si comprano, si portano a casa, o in ufficio, o a scuola, e *funzionano.* No, fortunatamente non è così, fortunatamente è un prodotto imperfetto!

NEWSLETTER

La Synertek seconda fonte per lo Z8 della Zilog

E' stato annunciato recentemente un accordo per cui la Synertek di Santa Clara (California) produrrà lo Z8, il microcalcolatore "single chip" a 8 bit e 4 MHz della Zilog.
Secondo l'accordo, la Synertek produrrà tutte le versioni dello Z8, lo Z-UPC (Universal Peripheral Controller), e il sistema di sviluppo per lo Z8.
Il presidente della Synertek, Robert Schreiner, ha commentato: "Lo Z8 soddisfa molti nostri obbiettivi di lungo termine, perchè siamo convinti che sia il miglior microcalcolatore monochip a 8 bit di alte prestazioni finora annunciato. Esso avrà vastissime applicazioni proprio nei campi di maggior interesse per la Synertek: calcolatori, periferiche di calcolatori e telecomunicazioni".
Lo Z8, con la SGS-ATES e la Synertek, ha ora due collaudate e credibili seconde fonti, che rafforzeranno il supporto di questo prodotto sul mercato mondiale.

Utilizzatori Apple, unitevi !

Si è costituita in Italia l'*Associazione Italiana Utilizzatori Apple*, similmente a quanto sta avvenendo in molti paesi europei ed in ogni grande città degli Stati Uniti.
Tali associazioni già hanno dato i loro frutti oltre oceano dove, funzionando in modo perfetto, non solo forniscono agli associati informazioni circa l'uso della macchina, ma promuovono scambi tali che ne fanno spesso rilevare la indispensabilità.
E' intenzione dell' *AIUA* pubblicare un news letter e promuovere incontri il primo dei quali è previsto per il prossimo dicembre in Roma.
Si riporta stralcio dello Statuto per quanto concerne le finalità dell'associazione:
Art. 3 - L'Associazione ha i seguenti scopi:
- promuovere la conoscenza e l'uso degli strumenti automatici di calcolo e di rappresentazione grafica sia di uso personale che collettivo;
- promuovere lo scambio di informazioni tecnico-scientifiche tra gli associati e tra l'associazione ed altri enti e singoli in particolare per quanto concerne il minicalcolatore denominato Apple.

Art. 4 - In particolare sarà compito dell'Associazione:
a) effettuare e promuovere incontri tra soci nel campo delle applicazioni dei sistemi automatici di calcolo e rappresentazione grafica;
b) promuovere l'approfondimento della conoscenza di tali sistemi;
c) effettuare e promuovere studi e ricerche sui problemi relativi ai sistemi suddetti;
d) agevolare e promuovere, anche mediante apposite attività editoriali, la diffusione e lo scambio di documentazioni, informazioni e studio, concernenti le materie di cui alle precedenti lettere a), b), e c), come pure la collaborazione, l'incontro, la discussione tra esperti, tecnici e studiosi interessati alla materia stessa;
e) dar vita e partecipare a qualsiasi iniziativa intesa ad analizzare i vari aspetti concernenti lo sviluppo e l'applicazione dei sistemi suddetti, a livelli sia locali che nazionali ed internazionali;
f) per il raggiungimento degli scopi sopra indicati l'Associazione potrà anche istituire Commissioni o Comitati di Studio a carattere temporaneo e conferire borse di studio.

La sede dell'AIUA è: Via del Boschetto, 29 00189 ROMA - Tel. (06) 4752277. Si invitano tutti coloro che sono interessati a prendere contatti immediati.

Uno studio Vantage sui micro

Fino a tutto il 1983 il mercato mondiale dei microcomputers su single board continuerà a svilupparsi ad un ritmo annuo del 40%. Lo afferma R. Wickman, presidente della Vantage Research Inc. di Mountain View, il quale aggiunge che la parte del leone continueranno a farla i sistemi da 8 bit. Per i microcomputers operanti su questa lunghezza si avrà nel prossimo quinquennio uno sviluppo medio annuale del 42% (in termini numerici) e del 22% (in valore). Per contro per i computers da 16 bit su una sola piastra si avranno tassi di crescita rispettivamente del 34 e del 33%. Nel 1978 il consumo mondiale di microcomputers su singola scheda sarebbe ammontato, secondo la Vantage, a 95.000 pezzi per un valore di 108 milioni di dollari di cui 53 milioni di dollari riferiti a dispositivi da 8 bit (60 mila pezzi) e 55 milioni di dollari riferiti a dispositivi da 16 bit. Quest'anno le previsioni formulate dagli esperti della Vantage Research assegnano ai microcomputers da 8 bit su una sola scheda un mercato di 110 mila unità (pari a 90 milioni di dollari) e a quelli da 16 bit un mercato di 55.000 pezzi (per un valore equivalente a 83 milioni di dollari).
A questa conclusione la casa americana è arrivata al termine di una survey nella quale sono stati presi in esame i tre tipici canali che alimentano il mercato dei singleboard computers:

a) quello delle società di semiconduttori (Intel, Motorola, Zilog, Mostek, National Semiconductor, etc.)
b) quello delle società di informatica impegnate in questo settore (DEC, HP, DG, CA, TI, etc.)
c) quello che nello studio viene chiamato canale dei fornitori indipendenti, come Pro-Log e Monolithics Systems.

Progetto Finalizzato Informatica

Su disposizione del CIPE il Consiglio Nazionale delle Ricerche ha avviato nel corso dell'estate 1979 il Progetto Finalizzato Informatica.
Entro la prima metà di Ottobre verrà consolidato, sulla base delle proposte e delle disponibilità per quella data manifestata, il programma esecutivo delle attività per il 1980.
Il progetto finalizzato di durata quinquennale, per il quale è stata prevista una disponibilità finanziaria complessiva di circa 25 miliardi, prevede la partecipazione di industrie e aziende private, di organismi della Pubblica Amministrazione, delle Università e del C.N.R. per l'espletamento di una coordinata azione di sviluppo avente come obiettivi fondamentali il rilancio dell'industria nazionale del settore, la qualificazione e l'ampliamento dei beni e servizi informatici e l'introduzione generalizzata di tecnologie informatiche nei processi industriali.
Sotto la direzione del Prof. Angelo Raffaele Meo del Politecnico di Torino, il Progetto si è articolato in tre sottoprogetti:
P1) Industria nazionale del settore: architettura e struttura dei sistemi di elaborazione.
P2) Informatizzazione della Pubblica Amministrazione.
P3) Automazione del lavoro e del controllo dei processi industriali.
La dimensione delle attività e dei settori oggi interessati alle applicazioni dell'informatica impongono al Progetto Finalizzato una funzione diversa da quella del gestore o del realizzatore: la funzione specifica del Progetto è quella "enzimatica" di punto di incontro e di strumento propulsore di uno sforzo di sviluppo coordinato cui tutti i partecipanti al Progetto devono sentirsi, ciascuno per i propri canali e nei propri ambiti, autonomi prosecutori.
In questa prospettiva è impegno del progetto contribuire al coordinamento degli strumenti di intervento pubblici, finanziari e di indirizzo, così come degli sforzi dell'industria nazionale, nelle sue componenti micro e mega aziendali, per coagulare ed amplificare la vitalità di un settore che nei prossimi anni sarà trainante per l'economia italiana.

Memorie a bolle magnetiche: accordo tra Siemens e Rockwell

La Siemens AG di Berlino/Monaco e la Rockwell hanno siglato un accordo per la produzione e la distribuzione delle memorie a bolle magnetiche prodotte dalle due società, compresi i relativi subsistemi. La Divisione componenti della Siemens AG produrrà la memoria a 256 K su licenza Rockwell. Entrambe le aziende distribuiranno in tutto il mondo anche le memorie a bolle magnetiche del proprio partner e si scambieranno per almeno cinque anni la tecnologia di questo specifico settore.

Un invito a «vivere l'avventura del personal computing».

La rivista Bit è redatta dai migliori esperti italiani della materia.
Per questo la fama di Bit ha ormai superato le nostre frontiere al punto che la rivista, paragonata alle migliori pubblicazioni straniere, ha instaurato con queste rapporti di stretta collaborazione da "pari a pari". Non stupitevi se qualche articolo di Bit appare su riviste americane, inglesi, francesi, tedesche, o se alcuni dei migliori articoli di queste appaiono su Bit.
E' normale che il "meglio" venga apprezzato e divulgato.
Ogni numero di Bit è un invito a "vivere l'avventura del personal computing" ma non solo questo ... software, hardware, analisi di investimenti, nuovi prodotti, business, finanza, contabilità, applicazioni didattiche, progetti personal e home, giochi, robotica. Migliaia di lettori italiani sanno tutto ciò. E voi?
Per assicurarsi gli otto numeri del 1980 è facile: basta abbonarsi.
Se volete anche i numeri arretrati scrivete alla nostra Redazione.
Accettate il nostro invito a "vivere l'avventura del personal computing", non ve ne pentirete.

Abbonarsi a

Bit

Certo ne vale la pena!

Arriva in Italia il personal della Apple

Fra i personal computers più noti e prestigiosi, l'Apple era ancora l'unico a non essere rappresentato in Italia: esiste infatti un accordo fra la casa americana e la ITT per la produzione in Europa di questo tipo di computer (anche se sembra doversi trattare di un modello in parte differente da quello commercializzato negli USA), per cui tutti gli occhi fino ad oggi sono stati puntati sulle mosse (mai fatte) della multinazionale.
A distogliere l'attenzione dal pachiderma che non dava segni di vita è stato l'annuncio non ancora ufficializzato fatto dalla IRET di Reggio Emilia in occasione di un breve colloquio avuto con il nostro giornale.
La IRET è una società specializzata nella distribuzione di apparecchiature per alta fedeltà che ha visto nel personal computer il nuovo business della propria strategia commerciale.
Giulio Bertellini, responsabile degli acquisti e promotore dell'iniziativa legata al personal computer, si è dichiarato estremamente soddisfatto degli accordi raggiunti con la Apple dopo una lunga e delicata trattativa: "Il nostro impegno sarà quello di mantenere l'immagine di gran prestigio che la Apple si è conquistata nel mondo sia dal punto di vista formale che per quanto riguarda l'assistenza tecnica e il software applicativo". "Inoltre - ha aggiunto - i prezzi saranno sorprendentemente interessanti".
Nei programmi della IRET c'è la nomina di vari distributori nelle più importanti città d'Italia e la creazione di centri di assistenza per un intervento rapido.

Grande successo Olivetti sul mercato scandinavo

E' stato firmato il 13 settembre a Ivrea il più grosso contratto di fornitura nella storia della Olivetti. Esso prevede la consegna di 6000 terminali (per un valore di circa 60 milioni di dollari) destinati all'automazione dei servizi di agenzia delle oltre cento Casse di Risparmio danesi che costituiscono una delle più estese reti di servizi bancari di tutta la Scandinavia.
Rispetto a quanto previsto nella lettera di intenti, firmata e resa nota nel marzo scorso, la consistenza dell'ordine è passata da 4000 a 6000 terminali, con un incremento del 50%.
L'Ing. De Benedetti, firmatario dell'accordo per conto della Olivetti, ha rilevato "il particolare valore di un accordo stipulato con uno dei più importanti consorzi di Casse di Risparmio nel mondo, il più importante in Europa".
Il '79 sembra destinato a siglare il successo di De Benedetti.

La Siemens come "second source" per l'8086

La collaborazione della Siemens con la Intel si estende ora anche ai sistemi a 16 bit. Nell'ambito degli accordi stipulati, la Siemens produrrà la famiglia 8086 della Intel come "second source", inoltre continuerà a supportare tale famiglia con i relativi mezzi per lo sviluppo hardware e software.
L'accordo prevede che la Siemens riceva dalla Intel anche un pacchetto tecnologico comprendente le maschere. In cambio la Intel riceverà il know how attinente a determinati prodotti che la Siemens sviluppa per la famiglia 8086.

La Efcis secondo fornitore per l'MC 68000 della Motorola

La società francese EFCIS (una consociata della Thomson-CSF e del Commissariato dell'Energia Atomica) nell'ambito degli accordi esistenti con la Motorola ha da questa ottenuto i diritti per la produzione e la commercializzazione del microprocessore 16 bit MC 68000. Della Motorola la società francese attualmente produce i microprocessori MC 6800 e le memorie associate.
Le prime diffusioni sul silicio del 68000 sono state effettuate ed Austin (Texas) nella prima parte dell'anno. In Europa sono più di 150 i clienti che hanno cominciato degli studi sulla base di campionature del nuovo microprocessore da 16 bit.
Nell'annunciare questo accordo, André Borel, direttore generale per l'Europa della Motorola, ha detto:
"Un passo importante nella conquista del mercato mondiale per il microprocessore MC 68000 è stato l'annuncio dell'accordo di secondo fornitore con la Hitachi, all'inizio dell'anno. Il nuovo accordo con la EFCIS conferma la nostra volontà di offrire al mercato europeo e mondiale la migliore scelta di fornitori come completamento ai mezzi di supporto tecnico e di produzione del nostro centro europeo MOS di East Kilbride Scozia".

Le proposte abbonamento 1980

1)	Abbonamento 1980 a Elettronica Oggi	L. 29.500 anziché L. 36.000 (estero L. 41.500)
2)	Abbonamento 1980 a l'Elettronica	L. 14.000 anziché L. 15.400 (estero L. 20.000)
3)	Abbonamento 1980 a Bit (8 numeri)	L. 12.500 anziché L. 16.000 (estero L. 17.500)
4)	Abbonamento 1980 a gli Strumenti Musicali (9 numeri)	L. 13.500 anziché L. 18.000 (estero L. 19.000)
5)	Abbonamento 1980 a Elettronica Oggi + l'Elettronica	L. 42.000 anziché L. 51.400 (estero L. 59.000)
6)	Abbonamento 1980 a Elettronica Oggi + Bit	L. 40.500 anziché L. 52.000 (estero L. 57.000)
7)	Abbonamento 1980 a Elettronica Oggi + gli Strumenti Musicali	L. 41.500 anziché L. 54.000 (estero L. 58.000)
8)	Abbonamento 1980 a l'Elettronica + Bit	L. 24.500 anziché L. 31.400 (estero L. 35.000)
9)	Abbonamento 1980 a l'Elettronica + gli Strumenti Musicali	L. 25.500 anziché L. 33.400 (estero L. 36.000)
10)	Abbonamento 1980 a Bit + gli Strumenti Musicali	L. 25.000 anziché L. 34.000 (estero L.36.000)
11)	Abbonamento 1980 a Elettronica Oggi + l'Elettronica + Bit	L. 48.500 anziché L. 67.400 (estero L. 68.500)
12)	Abbonamento 1980 a l'Elettronica + Bit + gli Strumenti Musicali	L. 34.000 anziché L. 49.400 (estero L. 48.000)
13)	Abbonamento 1980 a Elettronica Oggi + Bit + gli Strumenti Musicali	L. 49.000 anziché L. 70.000 (estero L. 69.000)
14)	Abbonamento 1980 a Elettronica Oggi + l'Elettronica + gli Strumenti Musicali	L. 50.000 anziché L. 69.400 (estero L. 70.000)
15)	Abbonamento 1980 a Elettronica Oggi + l'Elettronica + Bit + gli Strumenti Musicali	L. 55.000 anziché L. 85.400 (estero L. 77.000)

Inoltre per tutto il 1980 sconto 10% su tutti i libri editi o distribuiti dalla Jackson Italiana Editrice.

E per chi si abbona ad almeno due riviste una offerta eccezionale...

Potenziata la famiglia INTELLEC

La linea INTELLEC di sistemi di sviluppo è stata recentemente accresciuta dalla Intel, nella fascia degli "in circuit emulators", per espandere la copertura a tutti i microprocessori della famiglia, oltre naturalmente alla fascia dei computers su piastra singola. Oltre all'ICE80-85, diventano così disponibili:
ICE86: di supporto ad 8086 e SBC 8612 (consiste di 3 schede da inserire nell'MDS di cui richiede almeno 64K byte di memoria e gestisce la indirizzabilità completa dell'8086 fino a 1M byte di memoria).
ICE89: consente all'utente di emulare tutti i single chips della Intel (8049, 8048, 8748, 8039, 8035, 8021, ecc.) in tempo reale offrendo un debugging di tipo simbolico.
multi-ICE: è un package di software (MDS350) che traduce in realtà un'idea molto innovativa, quella cioè di supportare il progettista nelle applicazioni di tipo multiprocessor.
Il multi-ICE software consente l'emulazione di sistemi che contengono due 8085 oppure un 8085 e un chip della famiglia 8048, come naturalmente di SBC con più di un microprocessore sulla piastra.

Microprocessori: una tecnologia conservatrice o progressista?

Continuando di questo passo nello sviluppo della tecnologia dei microprocessori, si rischia di penalizzare la capacità degli utilizzatori di utilizzare la tecnologia costringendo l'industria dei semiconduttori ad investimenti di miliardi di dollari senza una effettiva contropartita d'ordine pratico. I microprocessori sono già arrivati ad un livello di sofisticazione e di prezzo tale da bastare per molte delle future applicazioni. La gran parte dei microcomputers oggi disponibili offrono un throughput e possibilità non sfruttate. La rapidità del rinnovo tecnologico è stata ed è tale che all'utente si sottrae il tempo di pensare alla soluzione del suo problema nella dovuta maniera. Sarebbe cioè più opportuno concentrarsi sull'apprendimento e lo sfruttamento delle attuali tecnologie che continuare così freneticamente a scoprirne di nuove. Questo il pensiero del dr. W. Davidow, vice-presidente e direttore generale della Intel (divisione sistemi microcomputer), che ha precisato che per il fatto di pensarla in questo modo non si sente un "conservatore dell'industria dei microprocessori".

La Technitron distribuisce prodotti per il trattamento analogico dell'immagine

La Technitron ha assunto la distribuzione di prodotti costruiti dalla Space Optic Research Labs per il trattamento analogico dell'immagine. Si tratta, in particolare, di una gamma di componenti e di sistemi ottici per il processamento dell'immagine mediante ottiche di Fourier. Fra i campi di applicazione si possono ricordare i seguenti: riconoscimento di forme, intensificazione dell'immagine, holografia per diffrazione di Fraunhofer, soppressione del rumore, etc.
Un catalogo completo con la descrizione di tutti i prodotti disponibili, delle applicazioni e delle bibliografie sull'argomento è a disposizione di tutti gli interessati che contatteranno direttamente gli uffici Technitron di Roma e di Milano.

Guerra fra microprocessori da 8 bit e da 16 bit?

Sebbene almeno una dozzina di microprocessori da 16 bit siano ora disponibili, la loro diffusione, in termine di vendite, tende ad essere inferiore alle attese (attualmente il consumo di microprocessori da 16 bit equivale al 6% dell'intero mercato, nel 1983 si prevede salirà al 23%). Parecchi progettisti si interrogano sulla utilità di dispositivi con tanti bit. Stando ad una disamina pubblicata su Microcomputer Analysis (una pubblicazione curata dalla Mackintosh Consultants), l'esigenza di parole con una lunghezza di 16 bit non è ricorrente in un gran numero di applicazioni, ma trova riscontro soltanto in particolari circostanze. È, ad esempio, il caso dei mini. Poiché i mini eseguono un'ampia varietà di programmi, il software può trarre sovente vantaggio da parole di 16 o anche di 32 bit. I microprocessori, viceversa, vengono per lo più usati per eseguire solamente un programma, per cui il vantaggio della larghezza delle parole raramente ripaga. I microprocessori da 8 bit, tenuto anche conto delle minori esigenze in termini di buffers e di memoria, avrebbero un indice di adattabilità più alto.
In un altro punto dell'analisi viene affermato che, complice anche il limitato supporto di software e di documentazione ai 16 bit, i microprocessori da 8 bit continueranno indisturbati a dominare i medi-grossi volumi di applicazioni durante gli anni ottanta.

sconto 50% sui libri Jackson

1) AUDIO HANDBOOK
Un manuale di progettazione audio con discussioni particolareggiate e progetti completi.
L. 9.500 (Abb. L. 4.750)

2) MANUALE PRATICO DEL RIPARATORE RADIO TV
Un autentico strumento di lavoro per i radioteleriparatori.
L. 18.500 (Abb. L. 9.250)

3) SC/MP
Applicazioni e programmi di utilità generale sul microprocessore SC/MP
L. 9.500 (Abb. 4.750)

4) IL BUGBOOK V
Esperimenti introduttivi all'elettronica digitale, alla programmazione ed all'interfacciamento del microprocessore 8080A.
L. 19.000 (Abb. L. 9.500)

5) IL BUGBOOK VI
Completa la trattazione del Bugbook V
L. 19.000 (Abb. L. 9.500)

6) IL TIMER 555
Descrive circa 100 circuiti utilizzanti il Timer 555 e numerosi esperimenti.
L. 8.600 (Abb. L. 4.300)

7) IL BUGBOOK I
Esperimenti sui circuiti logici e di memoria, utilizzanti circuiti integrati TTL.
L. 18.000 (Abb. L. 9.000)

8) IL BUGBOOK II
Completa la trattazione del Bugbook I.
L. 18.000 (Abb. L. 9.000)

9) IL BUGBOOK IIa
Esperimenti di interfacciamento e trasmissione dati utilizzanti il ricevitore/trasmettitore universale asincrono (Uart) ed il Loop di corrente a 20 mA.
L. 4.500 (Abb. L. 2.250)

10) IL BUGBOOK III
Questo libro fornisce una parola definitiva sull'argomento "8080A" divenuto ormai un classico nella letteratura tecnica sui microprocessori.
L. 19.000 (Abb. L. 9.500)

11) LA PROGETTAZIONE DEI FILTRI ATTIVI CON ESPERIMENTI
Tutto quanto è necessario sapere sui filtri attivi con numerosi esempi pratici ed esperimenti.
L. 15.000 (Abb. L. 7.500)

12) LA PROGETTAZIONE DEGLI AMPLIFICATORI OPERAZIONALI CON ESPERIMENTI
Il libro spiega il funzionamento degli OP-AMP, ne illustra alcune applicazioni pratiche e fornisce numerosi esperimenti.
L. 15.000 (Abb. L. 7.500)

13) CORSO DI ELETTRONICA FONDAMENTALE CON ESPERIMENTI
Un libro per chi vuole imparare partendo da zero.
L. 15.000 (Abb. L.7.500)

14) AUDIO & HI FI
Tutto quello che occorre sapere sull'argomento specifico.
L. 6.000 (Abb. L. 3.000)

15) COMPRENDERE L'ELETTRONICA A STATO SOLIDO
Dall'atomo ai circuiti integrati in una forma veramente didattica.
L. 14.000 (Abb. L. 7.000)

16) INTRODUZIONE PRATICA ALL'IMPIEGO DEI CIRCUITI INTEGRATI DIGITALI
Cosa sono e come si usano i CI digitali.
L. 7.000 (Abb. L. 3.500)

17) LESSICO DEI MICROPROCESSORI
Tutte le definizioni relative ai microprocessori.
L. 3.200 (Abb. L. 1.600)

18) INTRODUZIONE AL PERSONAL E BUSINESS COMPUTING
Il primo libro che chiarisce tutti i "misteri" dei personal e business computers.
L. 14.000 (Abb. L. 7.000)

19) LA PROGETTAZIONE DEI CIRCUITI PLL CON ESPERIMENTI
Teoria applicazioni ed esperimenti con i circuiti "Phase Loched Loop".
L. 14.000 (Abb. L. 7.000)

potete risparmiare
più del costo dell'abbonamento

TAGLIANDO D'ORDINE OFFERTA SPECIALE LIBRI SCONTO 50% RISERVATA AGLI ABBONATI AD ALMENO DUE RIVISTE JACKSON.

Da inviare a Jackson Italiana Editrice - P.le Massari 22 - 20125 MILANO

Nome ____________________
Cognome ____________________
Via ____________________ N° ________
Città ____________________ CAP ________
Codice Fiscale (indispensabile per aziende) ____________________
Data ____________ Firma ____________________

Inviatemi i seguenti libri:
(sbarrare il numero che interessa) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

☐ Pagherò al postino il prezzo indicato nella vostra offerta speciale + spese di spedizione
☐ Allegno assegno n- .... di L. ....
(in questo caso la spedizione è gratuita)
☐ Mi sono abbonato a: ☐ Elettronica Oggi ☐ L'Elettronica ☐ Bit ☐ gli Strumenti Musicali
a mezzo: ☐ Conto corrente postale ☐ Ho inviato assegno ☐ Presso il negozio ....
....

Questa offerta eccezionale é **valida solo fino al 31-1-1980 per un massimo di 5 libri** fra quelli elencati.

Per l'ordinazione compilate in ogni sua parte il tagliando inserito in questa pagina.

L'HP 64000: un potente sistema di aiuto allo sviluppo per microprocessori

Il nuovo sistema di sviluppo general-purpose per microprocessori che la Hewlett-Packard inizia a fornire in queste settimane costa all'utente una bella cifra: da un minimo di 25 mila dollari ad un massimo di 127 mila dollari. Inizialmente supporterà quattro fra i più diffusi microprocessori (8080 e 8085 della Intel, 6800 della Motorola e lo Z80 della Zilog); con l'estate del 1980 sarà possibile programmare anche altri microcomputers, compresi quelli da 16 bit. Il sistema, che si chiama HP 64000, non si limita a partorire programmi applicativi, ma educa altresì i programmatori ai sistemi operativi e ai linguaggi di alto livello. Il 64000 può supportare simultaneamente fino a sei diverse stazioni di sviluppo e, grazie all'impiego di un hard-disk come parte integrante dello stesso sistema, offre una capacità massima di 120 megabytes e veloci tempi di risposta.

La Nixdorf nel mercato del personal computer

Con l'acquisizione dei diritti di produzione e di commercializzazione su base mondiale del Lexicon LK 3000, un minitraduttore di basso costo sviluppato dalla Lexicon Corporation di Miami, la Nixdorf è entrata sul mercato dei personal computers. Per la esclusività della licenza la Nixford ha pagato circa 2,5 milioni di dollari, inoltre verserà delle royalties su ogni unità che venderà. L'LK 3000 è un calcolatore tascabile di tipo general-purpose ideato per funzioni di traduzione.
Fino ad oggi la Lexicon afferma di averne venduto circa 30 mila esemplari, ciascuno dei quali dotato di uno o più moduli intercambiabili per le seguenti lingue: inglese, francese, tedesco, italiano, greco, spagnolo, svedese e polacco. L'apparecchio ha le dimensioni e le caratteristiche di una calcolatrice tascabile. Viene venduto negli USA a 200 dollari. Ha una capacità di circa mille caratteri.
La Nixdorf programma di elevare il numero di moduli ad una quindicina e di farne una grossa promozione in occasione dei prossimi giochi olimpici di Mosca. Inoltre la casa tedesca ha in progetto di ampliare il ventaglio delle applicazioni trasformando l'LK 3000 in una stazione di lavoro per il data entry e sviluppando dispositivi I/O per l'interfacciamento con la televisione. La Nixdorf inoltre introdurrà dei moduli programmabili da parte del consumatore così da poter essere aggiornati.

Fra i progetti della Commodore trasduttori, sintetizzatori e calcolatrici con display

La maggior parte dei prodotti consumer che vedranno la luce nel corso degli anni ottanta utilizzeranno la tecnologia dei cristalli liquidi. Questa la motivazione in base alla quale la Commodore Business Machine (l'azienda dei PET) ha deciso di acquistare la Microdisplay Systems, un piccolo produttore americano di LCD. Dopo i microcomputers della serie PET, la CBM sta lavorando a parecchie altre nuove idee. Tra le tante spiccano questi progetti: un minitraduttore tascabile per il mercato turistico, un sintetizzatore vocale da inserire in apparecchiature domestiche e una calcolatrice del tipo hand-held basata su microprocessore e con un display alfanumerico.

Accordo incrociato fra Rockwell e Motorola

Rockwell e Motorola hanno convenuto di darsi una mano nel settore tecnologico. La prima cederà il know-how per costruire le bolle magnetiche da 256 K ed il board da 1 megabyte. Come ricompensa riceverà dalla Motorola le maschere e quant'altro necessario per realizzare i microprocessori 16-bit della serie 68000. La Rockwell, prima seconda fonte americana del potente dispositivo a 16 bit sviluppato dalla Motorola, prevede di iniziare le campionature a partire dal secondo trimestre del 1980.
Il 68000, ha dichiarato un portavoce dell'azienda, è stato selezionato perché considerato idoneo a "salvaguardare" gli investimenti fatti nel Super 65 in termini di risorse e di tempo nonché idoneo a complementarsi con la linea di micro 6500.

R6500 ADVANCED INTERACTIVE MICROCOMPUTER
Rockwell
AIM 65
Interactive Microcomputer
R6500
HARDWARE MANUAL
AIM 65
USER'S
R6500
PROGRAMMING MANUAL
AIM 65
MONITOR PROGRAM LISTING
AIM 65
AIM 65

Computer grafica

di **M. Montedoro**

Le capacità grafiche di un computer rappresentano uno degli elementi più affascinanti. In che cosa consistono le possibilità grafiche di un personal computer? Flash su PET, TRS-80, Apple II, Compucolor II.

Introduzione

I grossi calcolatori sono stati e sono tuttora impiegati per creare "disegni". Si può dire che tutti noi abbiamo avuto modo di vedere almeno uno dei moltissimi disegni realizzati stampando caratteri alfanumerici secondo un determinato modello; ad esempio Charlie Brown e Snoppy (v. Figura 1), che sono fra i soggetti a più grande diffusione.
Realizzare queste figure è semplice. Basta avere un microcalcolatore, una stampante e molta pazienza. Ciascuna riga di lettere è contenuta in un'istruzione di un programma scritto in BASIC. Si ha la stampa del disegno battendo il comando RUN. Ad un primo approccio si può usare un solo carattere, in genere l'X o lo 0, ripetuto molte volte per costruire il disegno, ma acquistando pratica si può ricorrere a versioni più sofisticate in cui ci si serve di caratteri diversi per rendere il chiaro-scuro.

Un altro modo per ottenere disegni è quello, recente, di fornire al microcalcolatore mediante una telecamera l'input per realizzare il "ritratto" (v. Figura 2). Il segnale proveniente dalla telecamera viene digitalizzato, e come tale è acquisito dal microcalcolatore; l'immagine in realtà è vista come formata da una serie di rettangolini, per cui il segnale digitale indica al microcalcolatore il grado di chiaro e di scuro di ciascun rettangolino. A questo punto il microcalcolatore associa al codice così fornito un carattere appropriato, che viene stampato in corrispondenza del rettangolino a cui il codice si riferisce.

In questi casi non si può parlare esattamente di computer grafica, nell'ambito della quale si evidenziano da una parte dispositivi di ingresso dati particolari (digitalizzatori di vari tipi oltre a tastiera) e altrettanto opportuni dispositivi di uscita (video grafici e plotters per uscite hard copy), dall'altra una struttura

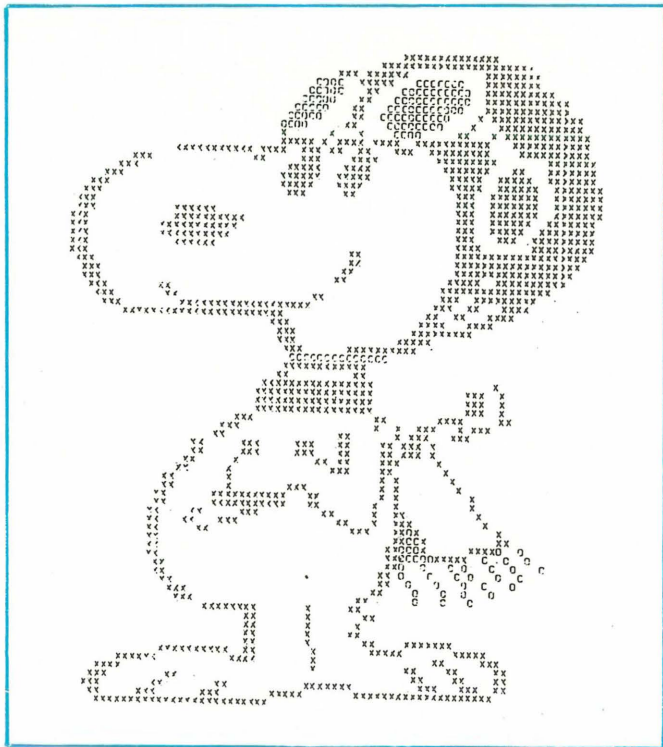



Figura 1 - Chi é?

hardware-software la cui finalità è quella di permettere la massima interazione utente-computer, in cui quanto è visualizzato è insieme il punto di partenza e di arrivo di tale attività. In quest'ambito le risorse software richieste possono essere molto complesse e comunque molto diversificate a seconda delle applicazioni (v. Figura 3), che vanno dalla semplice interpolazione di dati per generare curve o istogrammi (applicazioni statistiche), alla costruzione di disegni realizzati mediante il collegamento di forme elementari standard prelevate dagli archivi dati con la tecnica del menu, alla generazione di figure geometriche bidimensionali (disegni di impianti) o tridimensionali (caso della progettazione meccanica), in cui, dopo aver definito le entità geometriche piane o spaziali su cui si vuole operare, vengono create le varie figure tridimensionali.

Queste caratteristiche, proprie dei sistemi grafici, chiaramente comportano costi dell'ordine delle decine di milioni. Con un personal computer non bisogna aspettarsi prestazioni simili; comunque in una certa misura si possono avere prestazioni grafiche.

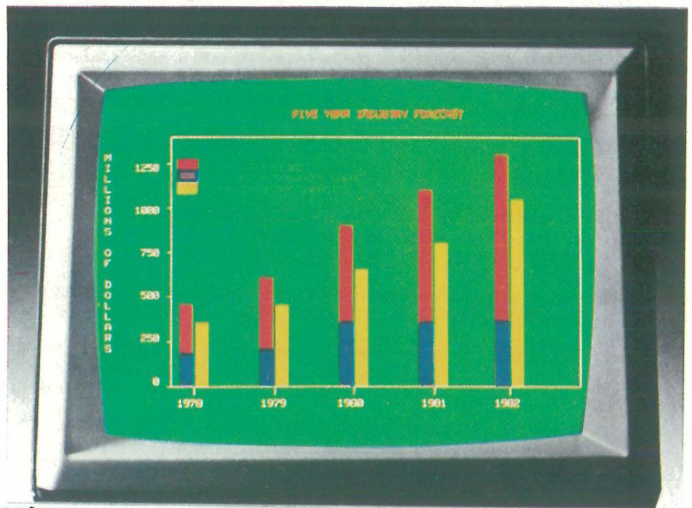
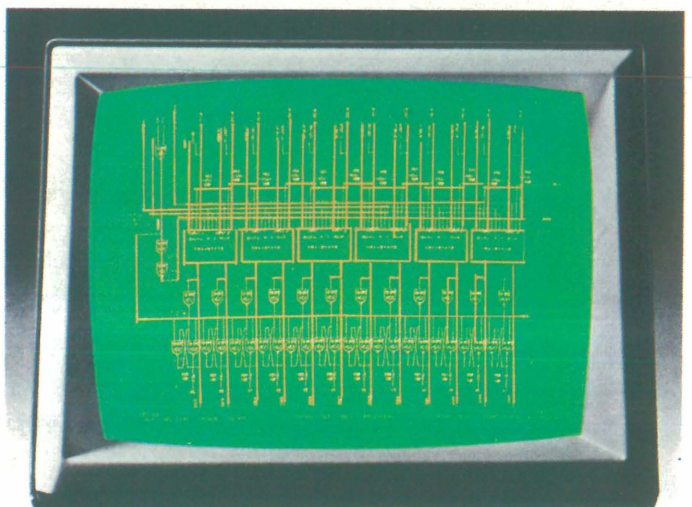
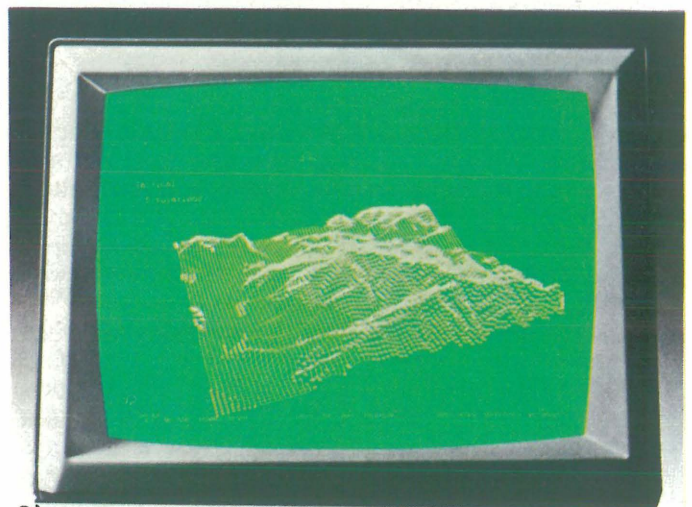
Figura 2 - "Ritratto" con telecamera e microcomputer.

a)

b)

c)

Figura 3 - Alcune applicazioni di computer grafica.
a) analisi dati; b) progettazione elettronica;
c) rappresentazione topografica.

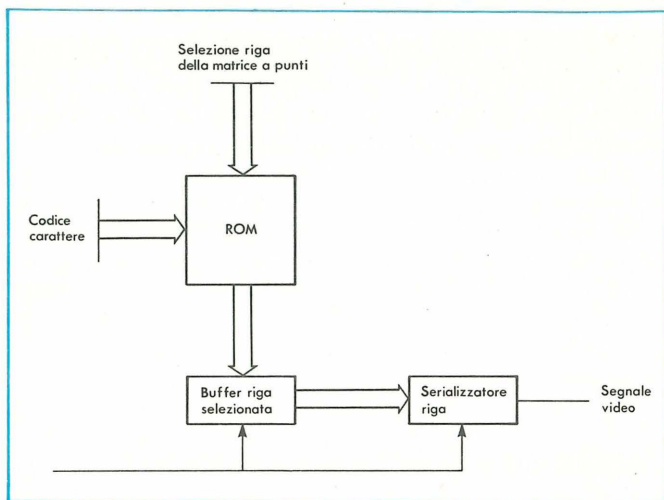



Figura 4 - Schema a blocchi di un generatore di carattere.

Diamo, perciò, alcune nozioni di base sui vari elementi che caratterizzano un sistema per applicazioni grafiche.

Video grafici

La visualizzazione dell'immagine è realizzata in due modi:
1) Mediante un normale tubo catodico (schermo TV), su cui l'immagine è inviata continuamente almeno 30 volte al secondo, in modo che l'occhio umano non percepisce nessuna discontinuità.
Questo metodo, detto a *rinfresco,* richiede che parte della memoria sia dedicata a contenere l'immagine di quanto appare sullo schermo. Più esattamente, ogni punto dello schermo rappresenta lo stato di un bit della memoria di video: se il bit è 1, il punto è luminoso, se è 0 il corrispondente punto è buio, in quanto i singoli bit sono usati per generare il segnale video. Cambiando la configurazione dei bit in memoria, mediante le istruzioni del computer, cambia l'immagine sullo schermo. Ad esempio l'animazione è ottenuta cambiando ad appropriati intervalli di tempo la configurazione di quella parte della memoria che si vuole animare. In definitiva questo modo richiede una memoria dedicata al rinfresco dello schermo tanto maggiore quanto più alta è la *risoluzione* con cui si vuole avere l'immagine nello schermo, e già questo è una differenziazione rispetto ad un video alfanumerico, in cui la risoluzione non è un fattore eccessivamente importante. Per un video grafico una bassa risoluzione comporta un'immagine più granulosa, essendo più piccolo il numero di elementi che formano l'immagine.
2) Il secondo modo consiste nell'utilizzo di uno schermo a memoria, cioè un tubo catodico che non ha bisogno del rinfresco dell'immagine in quanto è in grado di mantenerla per molto tempo. Premesso che tubi di questo tipo sono notevolmente più costosi di quelli normali anche se si considera il risparmio dovuto alla mancanza di memoria dedicata al rinfresco, è da sottolineare una diversità operativa rispetto ai sistemi con tubo normale, consistente nella necessità di pulire lo schermo ogni volta che occorre apportare delle variazioni all'immagine, che quindi è ritrasmessa integralmente. Se è innegabile il vantaggio di non essere costretti ad inviare ciclicamente l'immagine, in quanto ovviamente si risparmia tempo di CPU, questo tipo di video comporta delle limitazioni funzionali rispetto ad un sistema con video normale: ad esempio non sono possibili tecniche di animazione. Quanto si è detto finora chiarisce le ragioni della diversità di funzionalità e di prezzo dei sistemi per applicazioni grafiche. Nel caso dei personal computers si è cercato di trovare un compromesso tra le possibilità grafiche da fornire ed il prezzo, che si è cercato di mantenere ragionevolmente basso.

Figura 5 - Matrice a punti di un carattere nel formato 7 × 9.

Plotters

Fondamentalmente il plotter è una penna mobile pilotata dal computer. In un certo senso, i plotters sono l'equivalente hardcopy del video grafico, così come una stampante è l'equivalente hardcopy di un video alfanumerico.
I plotters, così come i video grafici, variano nella qualità della risoluzione. Molti plotters sono a coordinate cartesiane, il che significa che la superficie del disegno è divisa in un certo numero di trattini verticali ed orizzontali, e che il plotter può spostarsi lungo l'asse delle Y, lungo l'asse delle X e lungo una diagonale a 45 gradi (un trattino sull'asse delle X, e uno su quello delle Y.) La qualità del disegno finito dipende dalla lunghezza dei trattini che si sono usati. I sistemi più economici danno linee non continue, perché la penna va avanti a piccoli scatti. Così una linea curva risul-

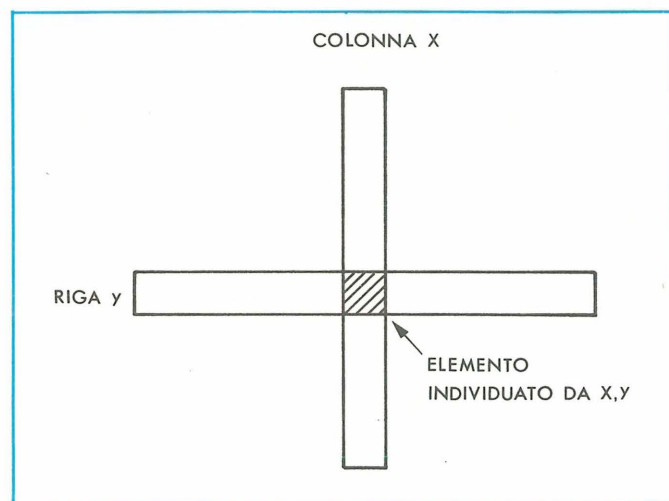



Figura 6 - Individuazione di un elemento dello schermo mediante coordinate cartesiane.

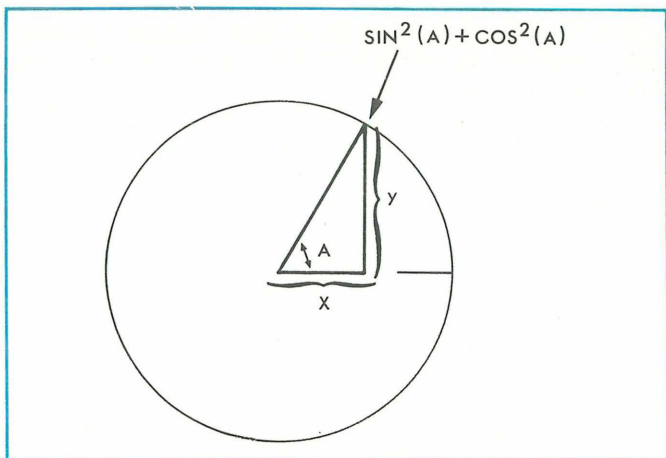



Figura 7 - Relazioni trigonometriche.

terà seghettata; apparirà continua se i trattini sono molto piccoli. E' da dire che molti dei plotters utilizzabili sui personal computers, tranne che nelle linee ortogonali e diagonali, segnano qualche irregolarità.

Altre periferiche

In ambito grafico vi sono dispositivi complementari finalizzati soprattutto a semplificare le operazioni di input dei dati grafici quando l'applicazione richiede una notevole attività di definizione degli stessi.

```
100 REM ****************************************
110 REM
120 REM    ROUTINE GENERALIZZATA PER
130 REM    GRAFICA GEOMETRICA
140 REM
145 REM ****************************************
150 REM
160 PRINT " FATTORE DI SCALA M = ";
170 INPUT M
180 PRINT " NUMERO DI PUNTI N = ";
190 INPUT N
200 PRINT " NUMERO DI CICLI S = ";
210 INPUT S
220 REM ************* DEFINIZIONE FUNZIONE *********
230 PRINT " PARAMETRO FUNZIONE P = ";
240 INPUT P
250 DEF FNR(A)=P*SIN(A)*COS(A)↑2
260 REM ************ GRAFICA ***********************
270 REM
280 PRINT " DIMENSIONI DI WINDOW : X1,Y1 = ";
290 INPUT X1,Y1
300 WINDOW -X1,X1,-Y1,Y1
310 VIEWPORT 30,130,0,100
320 PAGE
330 AXIS 1,1
340 MOVE -X1,Y1
350 DRAW X1,Y1
360 DRAW X1,-Y1
370 DRAW -X1,-Y1
380 DRAW -X1,Y1
390 MOVE 0,0
400 REM ************ ESECUZIONE DISEGNO ************
410 SET DEGREES
420 FOR I=1 TO N
430 LET T=360*S*I/N
440 DRAW M*FNR(T)*COS(T),M*FNR(T)*SIN(T)
450 NEXT I
460 HOME
470 PRINT USING 510:M
480 PRINT USING 520:N
490 PRINT USING 530:S
500 PRINT USING 540:X1,Y1
510 IMAGE "M=",3D.3D
520 IMAGE "N=",3D.3D
530 IMAGE "S=",3D.3D
540 IMAGE "X1,Y1=",3D," ",3D
550 END
```

Figura 8 - Programma di grafica geometrica.

Uno dei meno costosi è il *joystick*, che non è altro che una leva verticale che può essere mossa in una qualunque direzione. Il joystick è ideale per muovere un punto attraverso lo schermo televisivo. È un accessorio d'obbligo sugli home-hobby computers.
D'altra natura è invece il *tablet*, che non è altro che una superficie piana magnetica su cui viene mossa una "penna". La posizione della penna sul tablet è inviata al computer, che in tal modo è in grado di acquisire il contorno tracciato, ad esempio una mappa, un disegno, una fotografia. La risoluzione di questo accessorio è buona, potendosi rilevare dieci punti per millimetro.

Capacità grafiche del personal computers

Non tutti i personal computers sono dotati di video grafico nella configurazione standard, comunque anche per essi si parla di "capacità grafiche". In che senso?
Ad un primo livello s'intende che possono apparire lettere, numeri e *simboli speciali grafici* sullo schermo. Questi caratteri vengono visualizzati in un formato particolare, generalmente 16 linee × 64 caratteri, o 24 linee × 80 caratteri ciascuna. In questo caso non è valido quanto si è detto in precedenza, cioé la corrispondenza tra bit della memoria di rinfresco e punti sullo schermo. La corrispondenza è tra byte della memoria di rinfresco e simbolo (o carattere) visualizzato. La soluzione adottata è quella tipica degli schermi alfanumerici, in cui l'unità visualizzabile é il byte in cui é codificato il carattere (o il simbolo grafico da visualizzare).
In questo caso il byte é inviato ad un blocco generatore di carattere (v. Figura 4), che realizza la conversione da codice a dati seriali per il video. Questo blocco consiste fondamentalmente di una ROM (*R*ead *O*nly *M*emory), che contiene per ogni carattere la corrispondente configurazione in termini di punti luminosi (1) e non (0) disposti a matrice (tipicamente 5 x 7 oppure 7 x 9 come in Figura 5). La visualizzazione del carattere avviene inviando per ogni riga orizzontale dello schermo la corrispondente sequenza di bit della matrice (cioé la ROM di decodifica carattere é indirizzata dal carattere e dalla linea di scanning relativa alla linea di punti da visualizzare).
Questa soluzione é adottata ad esempio dal PET della Commodore, che ha uno schermo di 40 x 25, con una capacità totale di 1000 caratteri, ciascuno dei quali ha una matrice 8 x 8. Oltre ai 64 caratteri alfanumerici, il PET ha un set di 64 caratteri grafici, per un totale di 128 simboli, su ciascuno dei quali é possibile ottenere la configurazione complementare (cioé nero su bianco) accanto alla normale bianco su nero.

Nell'ambito dei personal che adottano la tecnica di considerare la memoria come sequenza di bit direttamente visualizzati sullo schermo, e che quindi considerano lo schermo come formato non da una successione di caratteri ma da un insieme di punti disposti a griglia, la bontà di questa soluzione é legata da un lato alla risoluzione, cioé al numero totale di punti in cui é suddiviso lo schermo, dall'altro al metodo usato per disegnare le immagini sullo schermo stesso. Ovvero, data la suddivisione dello schermo secondo una griglia più o meno fitta di punti, la di-

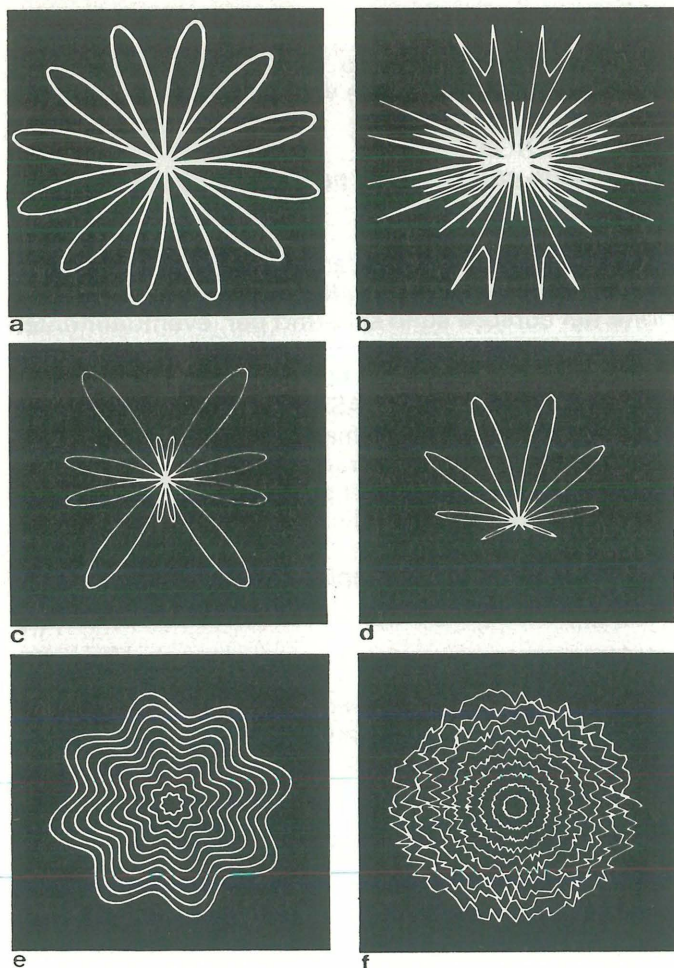



Figura 9 - "Disegni" generati dal programma di computer grafica.

Le funzioni utilizzate sono state:
a) SIN (12A);
b) SIN (A^2);
c) SIN (5A) COS (3A);
d) SIN (6A) COS (5A);
e) [(1+SIN[8A])/8]/1.25
f) [(1+RND[1])/5]/1.5.

stinzione nasce quando i singoli punti sono considerati dei "rettangolini" da rendere luminosi o bui, o invece dei punti da collegare fra loro con *vettori*. Questa seconda possibilità è chiaramente migliore della prima, ma anche più costosa.
Il TRS-80 adotta una soluzione a "rettangolini" disposti secondo una griglia di 128 colonne x 48 righe, per un totale di 6144 elementi, ciascuno dei quali é indirizzabile secondo le coordinate cartesiane X,Y, essendo la posizione (0,0) il primo rettangolino in alto a sinistra dello schermo. Tramite le istruzioni SET (X,Y) e RESET (X,Y) si comanda rispettivamente l'accensione e lo spegnimento dell'elemento indirizzato da (X.Y).

Questo metodo di indirizzare un punto secondo le coordinate è chiamato *plotting*, ed è usuale in ambito grafico (v. Figura 6). Altre istruzioni grafiche sono POINT (X, Y), che è usata accoppiata all'istruzione IF per determinare se il punto (X,Y) è acceso o spento, e PRINT AT NN "PIPPO", con la quale lo schermo é visto come formato da una sequenza di *caratteri* (non di rettangolini), in cui NN é la posizione del carattere in questa sequenza. Dato che lo schermo del TRS-80 ha una capacità di 16 linee x 64 caratteri, il massimo valore di NN é 1024. L'istruzione PRINT NN "PIPPO" scriverà nella posizione NN dello schermo il messaggio "PIPPO", e quindi serve in definitiva ad intercalare nel grafico visualizzato dei commenti leggibili. La memoria utilizzata per visualizzare una "schermata" di caratteri é uguale a quella grafica, dato che un carattere occupa 6 elementi grafici. Perciò in modalità grafica non é richiesta ulteriore memoria.

Caratteristiche grafiche più marcate possiede l'Apple II, che ha una capacità di testo di 24 linee x 40 caratteri. Ogni carattere può essere posto in qualunque posizione dello schermo e può essere visualizzato in modo normale o in modo complementare, o lampeggiante. Nella modalità grafica vera e propria può operare o in bassa risoluzione o in alta risoluzione. In bassa risoluzione l'Apple II ha una capacità di 40 x 40 elementi, con una finestrella sul fondo dello schermo per inserire righe di testo (massimo quattro), oppure di 40 x 48 elementi, senza finestrella di testo. In ogni caso ogni elemento può essere settato con un colore a scelta fra 16. Dal momento che la memoria dedicata al rinfresco dello schermo é la medesima di quella utilizzata nel caso alfanumerico, in questa modalità la memoria riservata é di circa 1 Kbyte.
In alta risoluzione l'Apple II permette solo quattro colori (nero, bianco, verde e viola), ma con una risoluzione di 280 x 192 punti (280 in orizzontale e 192 in verticale), per un totale di 53.760 punti, che é senz'altro una buona risoluzione, con la possibilità di aver quattro righe di testo in fondo allo schermo. In alta risoluzione comunque é richiesta come memoria di rinfresco almeno 8 Kbytes, per cui, mentre in bassa risoluzione il sistema minimo per lavorare necessita di soli 4 Kbytes di memoria, in alta risoluzione occorrono al minimo 12 Kbytes.

Della stessa classe dell'Apple II é il Compucolor II. Il video ha una capacità di 64 colonne x 32 righe, per un totale di 2048 caratteri. Ciascun carattere può essere rappresentato ad altezza normale o doppia e può essere colorato con un colore a scelta fra 8, così come lo sfondo. Accanto ai 64 caratteri alfanumerici ASCII, sono presenti 64 caratteri speciali (grafici), per un totale quindi di 128 caratteri. Ciascuno di questi caratteri é utilizzabile in modalità grafica in due modi:
1) Può essere posizionato in un punto qualunque dello schermo, e colorato.
2) Un carattere é considerato formato di 8 elementi, a ciascuno dei quali può essere attribuito un colore diverso. Il numero totale di elementi di griglia dello schermo é invece 128 x 128, per un totale di 16.384 punti, in quanto ogni carattere é scomposto in 8 elementi (2 in larghezza e 4 in altezza). Questa griglia di 16.384 elementi é quella a cui si fa riferimento nella modalità grafica vera e propria, con il generico elemento individuato dalle solite coordinate cartesiane.
Il Compucolor II lavora anche in modo vettoriale, e cioé, dopo aver indicato le coordinate (X_o,Y_o) e (X_1, Y_1) del vettore, é tracciata la linea che congiunge i due elementi estremi del vettore.

Istruzioni per la grafica

Abbiamo già accennato ad alcune di queste istruzioni. Al di là dei nomi, le funzioni svolte sono quelle

fondamentali per questo tipo di applicazione, e chiaramente rappresentano un'estensione del linguaggio BASIC. Comunque ora si farà riferimento ad un set di 7 comandi che implementano le funzioni basilari per il trattamento dell'immagine.
I comandi sono:
1) Comando di WINDOW
Formato: WINDOW X minimo, X massimo, Y minimo, Y massimo dove X,Y fanno riferimento ad un sistema di assi cartesiani. Questo comando permette di creare una scala per l'immagine. Supponiamo di voler creare un'immagine di dimensioni 10 cm. x 20 cm.; basterà usare il seguente comando:

WINDOW 0,10,0,20

Naturalmente le unità di misura di X ed Y saranno state preventivamente definite dall'utente o dal sistema.
Il comando di WINDOW in sostanza crea gli opportuni fattori di scala per far rientrare l'immagine nella *finestra* prevista.

2) Comando di AXIS
Formato: AXIS X intervallo, Y intervallo
Questo comando disegna, nello spazio fornito da WINDOW, gli assi X ed Y. I parametri del comando servono per stabilire la distanza a cui il sistema pone dei segni (trattini) sui due assi.
Ad esempio il comando

AXIS 5,13

fa sì che sull'asse X siano posti dei segni a 5, 10, 15, 20..., e sull'asse Y a 13, 26, 39....Il comando di AXIS é quindi usato per disegnare gli assi cartesiani con la suddivisione in intervalli regolari (coordinate). Le coordinate sono utilissime per la rappresentazione grafica di equazioni matematiche.

3) Comando di VIEWPORT
Formato: VIEWPORT start X, stop X, start Y, stop Y
Il comando serve per individuare la sezione fisica dello schermo per il posizionamento dell'immagine. Ad esempio, se lo schermo é una matrice di 130 x 100 punti, il comando

VIEWPORT 30, 130, 0, 100

fa sì che le immagini siano contenute in un rettangolo spostato di 30 unità rispetto all'angolo in basso a sinistra e con una base che coincide con l'ultima riga dello schermo (0).
Il comando

VIEWPORT 64, 65, 49, 51

fa sì che l'immagine sia posta in un rettangolino di dimensioni 2 x 1 collocato al centro dello schermo.

4) Comando di MOVE
Formato: MOVE X, Y
Fa sì che il cursore, senza disegnare nulla, venga mosso nella posizione X,Y. Il cursore può essere immaginato come la "punta della matita" del computer: con il comando di MOVE si sposta la punta nella pagina dello schermo tenendo la matita sollevata. Il cursore viene mosso dalla posizione in cui si trova alla posizione X,Y relativa all'angolo inferiore destro della WINDOW dentro VIEWPORT.

5) Comando di DRAW
Formato: DRAW X, Y
Ha lo stesso effetto di MOVE, solo che in questo caso la "punta" viene premuta sul foglio. Alternando MOVE e DRAW si possono costruire delle linee tratteggiate. DRAW disegna una linea retta che congiunge la posizione corrente del cursore con quella designata dai parametri. E' il comando fondamentale per disegnare delle linee (metodo vettoriale).

6) Comando di PAGE
Formato: PAGE
Permette di cancellare la pagina di schermo.

7) Comando di HOME
Formato: HOME
Posiziona il cursore nell'angolo superiore sinistro dello schermo. E' il comando che riinizializza la posizione del cursore sullo schermo per, eventualmente, stampare dei messaggi.
Oltre a questi 7 comandi base, altri comandi grafici sono spesso inclusi in BASIC, come: ROTATE, che permette di ruotare un'immagine; SET DEGREE, che permette di cambiare da radianti a gradi nel calcolo degli angoli. Nella maggior parte dei casi i 7 comandi base sono comunque sufficienti per ottenere prestazioni grafiche notevoli.
Ora vedremo alcune semplici ma suggestive applicazioni.

La nuova geometria

Dopo aver inizializzato lo schermo con i comandi di VIEWPORT (per individuare la posizione fisica) e di WINDOW (per la scala), se si vuol disegnare una figura, basta semplicemente scrivere una sequenza di comandi di DRAW: il computer farà il resto. In certi casi, durante l'inizializzazione, conviene introdurre il comando di AXIS.
Riportiamo ora un segmento di codice per rappresentare un rettangolo:

```
100 VIEWPORT    0, 130, 0, 100
110 WINDOW      0, 10, 0, 10
 120 MOVE       0, 0
 130 DRAW       0,10
 140 DRAW       10,10
 150 DRAW       10, 0
 160 DRAW       0,0
```

Supponendo che lo schermo sia una matrice 130 x 100, con l'istruzione n° 100 si alloca tutto lo schermo per la figura. Il comando di WINDOW riserva una finestra di 10 x 10, mentre il comando di MOVE posiziona il cursore nell'angolo in alto a sinistra della finestra. Le linee di codice da 130 a 160 tracciano i quattro lati del rettangolo.
Con le modalità riportate sopra si può rappresentare qualunque figura. Naturalmente volendo rappresentare delle linee curve bisognerà approssimarle con una spezzata: maggiore è il numero di segmenti della spezzata, maggiore sarà l'approssimazione.
Vediamo ora alcune nozioni base di trigonometria che possono servire per ottenere immagini veramente suggestive.

1) Funzione seno
La funzione seno (SIN) associa al valore di un angolo un numero variabile tra +1 e -1; più esattamente la funzione seno si azzera a 0°, 180° e 360°, mentre assume il valore assoluto massimo per angoli che valgono 90° e 270°.

2) Funzione coseno
La funzione coseno (COS) ha lo stesso andamento della funzione seno, solo che é anticipata: quando SIN é 0, COS é già 1.

Matematicamente la relazione tra seno e coseno é rappresentata da:

$$SIN (A) = -COS (A + 90)$$

cioé, dato un angolo che misura A°, il seno di quell'angolo é uguale - a meno del segno - al coseno di un angolo che é 90° più grande dell'angolo A.
Queste due funzioni ci permettono di definire in un modo più semplice (rispetto alla definizione con le coordinate (X,Y)) la circonferenza con raggio unitario come quella che soddisfa l'equazione:

$$1 = SIN^2(A) - COS^2(A)$$

Dalla figura 7 si vede che calcolando X e Y in base alle definizioni di seno e coseno, si ha:

X = raggio * COS (A)
Y = raggio * (A)

A questo punto ci sono sufficienti informazioni per disegnare un cerchio di raggio R:

```
500  FOR A = 0 TO 360 STEP 5
510  DRAW R*COS (A), R*SIN (A)
520  NEXT A
```

Queste tre fasi BASIC disegnano dei segmenti a 5° di intervallo attorno ad una circonferenza di raggio R. Tramite l'istruzione 500 si percorre tutta la circonferenza con incrementi di 5 gradi.

Variando le funzioni associate al comando di DRAW, é naturalmente possibile ottenere una gran varietà di figure. Supponiamo di voler disegnare una figura che é funzione dell'angolo A. Possiamo definire la funzione in un sottoprogramma BASIC di definizione funzione:

```
490  DEF FNR(A) = -EXP (P*A)
```

Associando questa linea di programma (in cui é utilizzata la funzione EXP) a quelle scritte precedentemente, si otterrà una spirale. In Figura 8 é riportato un programma in cui basta semplicemente cambiare il sottoprogramma di definizione della funzione per ottenere le immagini di Figura 9. Le funzioni richieste sono riportate nella didascalia.

Conclusioni

Il mondo del computer é veramente affascinante. Certo non può essere affrontato a digiuno di qualunque nozione. Con gli ultimi esempi si é voluto far vedere come con schemi semplici é possibile ottenere risultati affascinanti. Per restare nel campo dell'applicazione mostrata, e per non farci prendere la mano da considerazioni circa l'*arte* di creare immagini per mezzo di un computer, riteniamo opportuno citare un santone dell'informatica: M. A. Arbib: "...le immagini...benché generate dallo stesso programma, hanno uno "stile" diverso. Il computer fa questi disegni con un programma veramente semplice, eppure mostra una piacevole varietà di trame e forme. C'é gente che chiede se il computer può contribuire all'arte; la mia risposta é che lo può senz'altro."

Bibliografia

1) M.A.Arbib, *Computers and Cybernetic Society* (Accademic Press, 1977).
2) T.G. Lewis, *The Mind Appliance* (Hayden Book Company, Inc., 1978).

Computer grafica con un minisistema

di **M. Salvemini,** *Università dell'Aquila, Facoltà di Ingegneria.*

La computer grafica ha fatto il suo ingresso nel mondo dell'informatica e della comunicazione in generale.

La rivista *Computer Decisions*, September 1979, stima che durante quest'anno verranno installati nel mondo più di 150.000 terminali per computer grafica.
Il costo dell'immagine in questi ultimi tempi è calato notevolmente: quello che prima era prerogativa o possibilità dei grandi centri di calcolo e di uffici specializzati è oggi alla portata di tutti e tutti sembrano interessati a fruirne.
L'uso e lo sviluppo della computer grafica non è legato solamente alla diminuzione dei costi di hardware, ma anche e soprattutto ad un generale movimento della cultura tecnico-scientifica e di mass media che ha privilegiato la comunicazione visiva rispetto ad altri tipi di comunicazioni.
La computer grafica, in quanto aiuto meccanico per ottenere rapidamente ed a basso costo immagini dense di informazioni, è quindi subito apparsa come la tecnica da privilegiare per ottenere la comunicazione più potente.
L'apporto dei minisistemi e degli small e microcomputers è stato in questo sviluppo di importanza fondamentale, non solo perchè i computers hanno prezzi molto più accessibili di alcuni anni fa, ma perchè essi sono costruiti in maniera tale da facilitare l'approccio di persone non specializzate. Tale approccio avviene generalmente attraverso messaggi ad alto contenuto informativo: le immagini.
È quindi certo che ci sarà non solo lo sviluppo di micro e small computers dedicati alla grafica, ma anche il potenziamento delle capacità grafiche di quelli *general-purpose.*
La computer grafica in Italia non ha avuto sino ad oggi uno sviluppo intenso anche se in alcune situazioni esso è stato puntuale, finalizzato ed ha fornito ottimi risultati.
Nell'ambito dell'Università italiana ad esempio non esiste un impegno reale nei confronti della computer grafica, cosicché è possibile "monitorare", qua e là, alcune interessanti esperienze, ma non una reale scuola: alcune proposte in tal senso per la creazione di corsi ad hoc sono state fatte dalla Facoltà di Ingegneria dell'Università dell'Aquila e giacciono nelle more di decisioni non facili da prendere.
Nell'ambito dell'Industria è interessante notare che proprio al fine di modernizzare sistemi di lavorazione, rendere più agevoli processi di fabbricazione etc. esistono interessanti applicazioni della computer grafica e del *Computer Aided Design*.
Nell'ambito specifico dei miei interessi, che sono le applicazioni al territorio ed agli insediamenti umani in generale, la situazione in Italia è abbastanza povera.
Esistono spesso idee brillanti, ma purtroppo operativamente non esiste ancora un sistema informativo operante basato sulle immagini in grado di fornire informazioni sul territorio, sulla sua consistenza, le sue modificazioni etc.
Vari progetti sono in corso, alcuni in fase di avanzata realizzazione, e presto sarà possibile avviare la fase di valutazione dei risultati ottenuti.
Parlare di computer grafica è parlare di un campo ben più vasto delle applicazioni al territorio: per non entrare in dettagli che potrebbero essere anche tediosi, mi è sembrato opportuno affrontare tre temi e trattarli mediante esempi:
- Parole ed immagini
- Immagini contornate ed immagini colorate.
- Immagini piane ed immagini tridimensionali.

Parole ed immagini

Molto si è parlato del rapporto tra parole ed immagini: molti studi, anche complessi, sono stati compiuti circa la quantità di parole che è possibile trasferire tramite una immagine.
Senza scendere in particolari è possibile affermare che la migliore valutazione della convenienza dell'immagine nei confronti della parola proviene dal mondo dei business.
Il sistema audiovisivo, ad esempio è oramai diventato non solo complemento della presentazione e discussione del manager, ma addirittura l'aspetto al quale più si presta attenzione nella preparazione di una presentazione.
Di fatto i rapporti scritti di tutti i generi hanno raggiunto oramai lunghezze esasperanti e non è facile ed agevole leggerli: occorre fare i conti con il tipo di linguaggio usato, con la sua difficoltà; con il tipo di terminologia usata etc.
Le immagini invece sono fruibili immediatamente ed in maniera sintetica.
Questo non significa che tutte le immagini siano buone e convenienti. Occorre avere ben chiaro che cosa trasmettere per loro tramite e poi disegnarle in modo tale che la comunicazione dell'informazione sia assicurata.

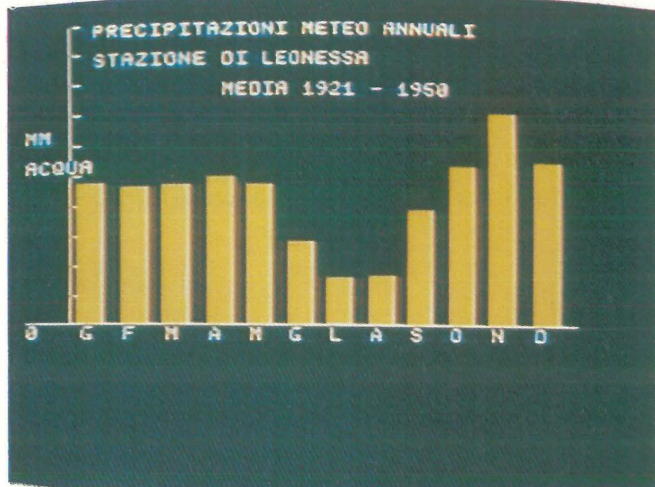



Figura 1 - Il diagramma barre é un tipico esempio di come alcune informazioni quantitative possano essere trasformate in immagini ed essere più facilmente interpretabili e confrontabili dei singoli numeri o delle tabelle.
La figura mostra le precipitazioni meteoriche misurate in mm di acqua per tutti i dodici mesi dell'anno in una stazione di rilevamento situata nella regione Lazio (nella cittadina di Leonessa). Il diagramma creato su schermo da un programma legge archivi dati scelti dall'operatore e li graficizza secondo scale ad hoc permettendo di avere informazioni quantitative sul singolo mese (quantità di acqua caduta) e di fare confronti: in questo caso il mese più piovoso é stato novembre, mediamente nell'arco di anni considerato, quello meno piovoso luglio.

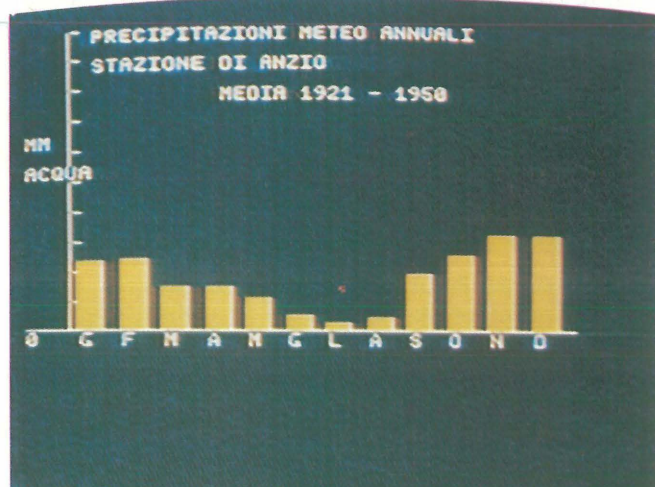



Figura 2 - La figura mostra l'output grafico del programma mostrato nella Figura 1, ma con un altro data file, relativo ad un'altra stazione di rilevamento (la cittadina di Anzio, sempre nel Lazio ma situata sulla costa). Non solo é possibile anche in questo caso ricavare informazioni quantitative relative alla pioggia caduta durante i vari mesi nella stazione, ma, riferendosi al diagramma mostrato nella Figura 1, è possibile (essendo le stesse le scale, le ascisse e le ordinate) compiere raffronti sia in totale che mese per mese.
Un esperto di climatologia potrebbe certo trarre interessanti considerazioni dai due diagrammi, che comunque potrebbero essere utili anche per i non esperti nella scelta....del periodo per una vacanza....mediamente parlando!!!!!!

Esistono molti esempi di immagini non utili e di difficile lettura: nell'esaminare le piante dei trasporti pubblici di varie città del mondo è facile trovarne di poco facilmente leggibili: purtroppo tra gli esempi peggiori è da annoverare quello di Roma.

Una volta assunto che è agevole e utile usare le immagini come mezzo di comunicazione, è necessario prendere delle decisioni sul come realizzarle e come trasmetterle.
Si può utilizzare un supporto di tipo rigido cartaceo oppure un supporto di tipo magnetico.
Usando quello cartaceo la trasmissione è immediata;

Figura 3 - Nella figura si rappresenta il sistema le cui specifiche sono riportate nel testo. L'ingombro del sistema stesso è molto modesto: tutti i componenti trovano posto su di un tavolo di media grandezza, escluso il televisore a colori.
Il monitor viene generalmente usato per lavorare. E' anche impiegato per i grafici in alta risoluzione per la nitidezza che altrimenti sarebbe difficile ottenere con TV color. Esso infatti é un normale televisore a colori di buona marca (che é possibile trasformare in monitor a colori). Specie per lavori di computer grafica é importante avere due disk drivers per la quantità di dati che generalmente si manipolano. Il *digitizer* (tavoletta magnetica) è molto utile per l'introduzione dei dati; è anche possibile fare manualmente il lavoro di immissione dati, che però così risulta più tedioso e soggetto a frequenti errori.
Sulla stampante poi, anche se solo alfanumerica, é possibile ottenere grafici e diagrammi di sicuro effetto ed utilità. In mancanza della stampante e comunque per avere una copia immediata di quanto mostrato sul televisore a colori, é conveniente avere a portata di mano una instant camera; il risultato non é sempre dei migliori, ma serve a fissare l'immagine.

Figura 4 - La figura mostra i confini delle venti circoscrizioni del Comune di Roma: essi sono stati digitalizzati e processati mediante un programma che permette di rappresentarli in toto o in parte.
La figura è un ottimo esempio di mappa realizzata mediante contorni.
E' possibile notare come individuare una circoscrizione dall'altra, senza che ci siano indicazioni di carattere alfanumerico, sia poco agevole. Tali indicazioni del resto non sono state poste sul disegno, al fine di rendere il disegno stesso più leggibile. Qualore l'utilizzatore voglia ottenere specifiche informazioni su una o più circoscrizioni, è opportuno usare il programma ZOOM: una subroutine in grado di ingrandire e mostrare in modo conforme le parti richieste ed i dati ad esse relativi.

Figura 5 - La figura mostra come il contenuto informativo della mappa precedente, realizzata solo per confini, aumenti considerevolmente qualora le aree vengano campite con colori funzione del valore raggiunto in ciascuna area del parametro e/o grandezza che si sta esaminando. In questo caso si è presa in considerazione la popolazione residente per circoscrizione. I colori sono proporzionali al valore della popolazione nella circoscrizione e vanno in ordine decrescente procedendo al bianco, al giallo, al nero, al blu, al viola, al verde.
A richiesta si può ottenere una tabella che mostri i valori raggiunti; é opportuno comunque notare come una mappa di tal genere possa fornire realmente una grande quantità di informazioni. Prima tra tutte, quella relativa alla densità di popolazione: infatti, poiché i colori sono allocati in aree, è possibile effettuare il confronto non solo tra i colori ma anche tra la superficie delle aree campite, la distanza di ciascuna area dal centro della città, etc.

di fatto è un disegno, diagramma etc, che può essere immediatamente allegato a relazioni, rapporti o fatto circolare, duplicato, etc. Usando il supporto di tipo magnetico invece occorre disporre di macchine che decodifichino le informazioni digitali e permettano di rappresentarle su devices (terminali) ad hoc.
Tali terminali sono generalmente monitor, a colori e non, che possono essere guidati mediante una tastiera e possono produrre una copia su supporto rigido di quello che è mostrato sullo schermo, qualora siano forniti di una attrezzatura chiamata hard copy in grado di ricopiare su carta (a colori e non) quanto appare sul terminale.
Per realizzare copie direttamente su carta da allegare a relazioni e/o pubblicazioni è possibile usare i plotters o anche le stampanti, che però producono disegni discreti, cioè non mediante tratti continui come i plotters.
Gli small e personal computers ricoprono un ruolo importante nell'ambito dell'elaborazione e trasferimento dell'immagine.
Mentre infatti le immagini elaborate su grandi sistemi richiedono reti di trasferimento, dati ed attrezzature complicate, il supporto magnetico in uso per lo small e personal computer è agevole da maneggiare, il computer stesso ed i suoi devices sono di piccole dimensioni ed agevoli da trasportare, e soprattutto possono essere dedicati a singole utilizzazioni con notevole rendimento dell'investimento effettuato.
Per quanto riguarda la collocazione funzionale di tali apparecchiature, esse appaiono particolarmente adatte a trovare posto sulla scrivania di che deve prendere decisioni ma non è in grado di elaborare i dati dai quali invece gli è utile e indispensabile estrarre rapidamente informazioni sintetiche e comprensibili.
È ovvio che per quanto concerne vasti archivi di dati non è possibile utilizzare minielaboratori, specie se l'elaborazione deve avvenire in tempo reale: è anche vero però che il piccolo elaboratore è in grado di dedicarsi, in tal caso, alla sola graficizzazione dei dati una volta che essi siano stati processati da macchine con più ampie disponibilità di memoria.

Figura 6 - Si mostra qui il processo di campitura di ciascuna area che viene svolto dalla macchina presenti i confini delle circoscrizioni. Questo è particolarmente importante perché permette di colorare anche solo una circoscrizione sulla quale eventualmente si vuole richiamare l'attenzione e/o della quale si vuole studiare un fenomeno in particolare.

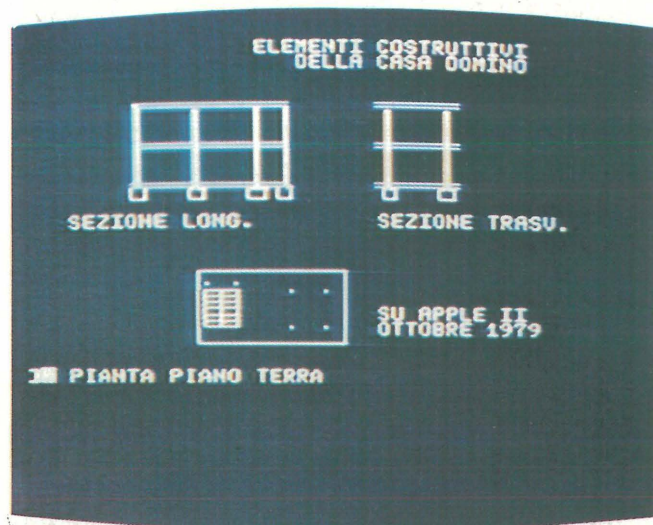



Figura 7 - La computer grafica ha trovato uno dei campi di maggiore applicazione nelle costruzioni ed in genere in tutte le fasi progettuali, anche quelle di verifica. Qui é esaminata dettagliatamente un'opera architettonica (del resto mai realizzata): la casa Domino di Le Corbusier, la prima realizzazione in cemento armato che assicurava il piano completamente libero dalle murature di sostegno, affidando tutto a strutture puntiformi (i pilastri).
La figura mostra le tipiche sezioni che corredano generalmente un progetto: quella trasversale, quella longitudinale e la pianta.
Questo tipo di rappresentazione é sicuramente efficace ed esaustivo anche delle necessità di comunicazione tecnica, ma in ben altro modo rappresentano l'oggetto le immagini successive, che con lo stesso data file usano invece un sistema di rappresentazione a 3 dimensioni.

Immagini contornate e immagini colorate

Uno degli esercizi più comuni nella tecnica del *pattern recognition* è quello del riconoscimento degli oggetti tramite la loro forma non conoscendone il contenuto, cioé quello che giace all'interno del contorno.
Molte volte in computer grafica è opportuno utilizzare le *contour lines*, cioé le linee che permettono di contornare un oggetto denunciandone la forma; esse permettono il calcolo delle aree, dei perimetri e delle sezioni e consentono ad un operatore specializzato di individuare quanto è contenuto all'interno delle aree e dei contorni.
Esempio nella disciplina del *pattern recognition* è il riconoscimento dei carri armati mediante foto aeree: la forma tipicamente rettangolare di un ambiente non dotato di forme geometriche semplici e schematiche permette l'individuazione di quanto ricercato.

Più difficile da registrare è la situazione quando, oltre all'eventuale carro armato, la foto aerea mostra (in quanto ripresa non in condizioni azimutali e/o con particolari condizioni di luce) anche l'ombra del carro: le dimensioni in tal caso non rispondono più alle caratteristiche note, gli angoli risultano smussati ed in genere il fotointerprete trova maggiori difficoltà per individuare il pattern.
Tutto questo serve a spiegare che alcuni oggetti e forme possono essere individuati solamente tramite i contorni; è necessario però che l'osservatore sia abituato alle forme che tenta di interpretare, altrimenti si possono incontrare serie difficoltà.

Un esempio tipico è quello riportato con riferimento al Comune di Roma e alla suddivisione in circoscrizioni del territorio comunale.
Le immagini di figura 1 mostrano la pianta del comune di Roma ed i confini delle sue venti circoscrizioni: essi sono stati digitalizzati e memorizzati, quindi le subroutine preparate ad hoc hanno permesso di elaborare i dati delle immagini in relazione ai dati relativi a ciascuna delle circoscrizioni.
Non per tutti è agevole capire a quale circoscrizione appartiene ciascuno dei contorni mappati; ben diverso invece sarebbe se si potesse disporre di una pianta nella quale ogni circoscrizione fosse colorata con un colore diverso, cosicchè si potesse dire: "Io abito nella rossa", "io abito nella verde" e così via.
Il colore è un codice ed il suo uso facilita la trasmissione di informazioni, in particolare di quelle planimetriche.
Le immagini colorate quindi sono più potenti delle immagini contornate dal punto di vista della quantità di informazione trasmessa.
È d'altra parte da rilevare che un'immagine colorata divisa in più parti colorate nello stesso modo non fornisce indicazioni sui confini delle singole aree.
Il maggior contenuto informativo è quindi proprio di un'immagine che usi i contorni per individuare ciascuna area ed i colori per individuarne le caratteristiche qualitative e quantitative.

Immagini piane ed immagini tridimensionali

Gli oggetti nello spazio reale sono fruiti dall'uomo come immagini tridimensionali.
Per semplicità nella trasmissione di informazioni, specie tra specialisti, usiamo immagini bidimensionali che sono più facilmente disegnabili e trasferibili; esse vengono generalmente chiamate piante, sezioni, prospetti.
Il problema principale di tali immagini è che esse sono difficilmente fruibili da chi non possiede i codici di lettura.
È perciò difficile che una qualunque persona, posta davanti al disegno della propria abitazione, la riconosca, mentre è molto più facile che riconosca una delle stanze nelle quali abitualmente vive guardando un suo disegno prospettico o ancora meglio, osservando una fotografia scattata all'interno della stanza stessa.
L'immagine fotografica è senz'altro il sistema migliore per rappresentare su carta un oggetto a tre dimensioni.
Il problema è che non è sempre possibile scattare fotografie degli oggetti tridimensionali: ciò si realizza sicuramente quando essi non esistono.
Il caso tipico a tal proposito è quello della progettazione: in essa il progettista immagina un oggetto, lo graficizza su carta usando codici ed a quel punto vorrebbe che fosse realizzato per poterlo toccare, vedere da tutte le parti, entrarci dentro se è grande abbastanza, accostarlo ad altri oggetti per vedere se si adatta.Fare tutto ciò o è difficile e complicato, oppure impossibile se non a costi terribilmente alti.Specie nella costruzione di un edificio non è certo possibile realizzarne alcuni per prova e poi scegliere il definitivo.Generalmente si elimina questo inconveniente facendo modelli, plastici in scala; guardandoli attentamente, fotografandoli etc. È possibile avere un'idea abbastanza certa di che cosa si è progettato.Anche questo generalmente è dispendioso e poi non permette di entrare dentro il modello (nel caso di un edificio) per poter fruire lo spazio architettonico creato.
Esistono sistemi molto sofisticati per simulare la fruizione di spazi tridimensionali ed in genere di visioni create artificialmente.Tra questi quello messo a punto da alcuni ricercatori americani che permette di proiettare tramite un opportuno occhiale direttamente sulla pupilla un'immagine, in modo che il fruitore, non vedendo nient'altro che ciò che gli viene proiettato sulle pupille, si trovi di volta in volta nello spazio artificialmente creato sotto forma di immagine.

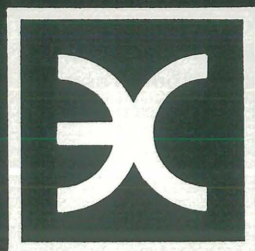
Un sistema meno sofisticato ma certamente più alla portata del pubblico è quello di rappresentare mediante restituzione prospettica lo spazio a tre dimensioni. Questo è possibile anche usando mini e micro computers.

Si tratta generalmente di digitalizzare quanto disegnato su una sezione, di fissare il punto di vista dell'osservatore e di far disegnare alla macchina quanto dall'osservatore dovrebbe essere visto dal punto di vista assegnato e nella direzione predisposta.

Esiste anche la possibilità di utilizzare la terza dimensione su oggetti ed immagini bidimensionali per fornire ulteriori informazioni all'osservatore, informazioni generalmente di carattere quantitativo.

È questo il caso delle mappe e carte urbanistiche realizzate in 3 dimensioni, dove la terza dimensione offre la possibilità di fornire informazioni quantitative per ciascun punto per il quale viene riportata.
Questo sistema di graficizzazione viene da alcuni cri-

Ediconsult la rivoluzione dei microcomputer.

EDIconsult dice: "BASTA AGLI ELABORATORI DI ALTO PREZZO" I microcomputer stanno dando una scossa decisiva al mercato E.D.P. Alla base della rivoluzione dei microcomputers sono i microprecessori. La loro tecnologia,modernissima,validissima,di basso costo è alla portata dei piccoli produttori e determina il loro inserimento nel mercato e l'abbattimento dei prezzi. E' bene che oggi l'utente sappia che, a fronte dei costi delle grandi case monopolizzatrici,determinati dagli sprechi delle loro strutture smisurate, sono disponibili, alla portata di qualsiasi azienda, - microelaboratori personali a prezzo inferiore a 1,5 milioni - microelaboratori per applicazioni gestionali o dedicate, completi di 32 K di memoria RAM, videomonitor e due floppy dischi a prezzo inferiore di 7 milioni - microelaboratori per applicazioni gestionali o dedicate di notevole mole, completi di 60 K di memoria RAM, videoterminale e dischi grandi (decine di milioni di caratteri) a prezzi inferiori a 20 milioni - software di base semplice, completo e potente incluso nel prezzo - procedure applicative standard gestionali a prezzo inferiore a 2 milioni. Questi validissimi microcomputers sono i migliori esistenti oggi sul mercato, dotati di completi programmi applicativi di utilizzo facile anche all'operatore inesperto, di semplice manutenzione. EDIconsult li offre a una cifra incredibilmente bassa rispetto a quella delle grandi case, perchè fa pagare solo il valore industriale di produzione e vendita delle apparecchiature e l'assistenza, solamente quando il cliente le richiede. Collaborate con EDIconsult alla diffusione degli elaboratori a basso costo; EDIconsult é garanzia di alto valore a basso prezzo

EDIconsult

S R L, Via Rosmini 3, MONZA Tel. 039 389.850 - 360.727

ticato in quanto non comprensibile. È da osservare che risulta a volte poco comprensibile se si sceglie un cattivo punto di vista, ma la difficoltà non è intrinseca all'immagine quanto al problema della mancanza di possibilità da parte dell'osservatore di fruire l'immagine di grandi spazi.
I problemi sono simili a quelli che insorgono nel riconoscimento delle caratteristiche fisiche del territorio tramite le foto aeree da parte di persone non specializzate.

Un minisistema per computer grafica

Da alcuni mesi è stato messo a punto ed è operante un minisistema dedicato alle applicazioni della computer grafica ed in generale a processare dati di varia natura con la finalità di trasformare i risultati sia numerici che funzionali in grafici.
Lo schema dell'organizzazione di tale sistema è mostrato in Figura, ed i suoi componenti specifici sono:

- personal computer da 48 K "Apple 2" dotato di possibilità di interfaccia con varie devices e programmabile in BASIC e Pascal;
- due disk drivers per dischetti da circa 100 Kbytes e relativa interfaccia con il computers;
- stampante alfanumerica ed interfaccia parallela di collegamento;
- digitizer con risoluzione di 200 linee/inch e con possibilità di trasmissione di 100 paia di coordinate per secondo;
- ROM e cards preparate ad hoc, di aiuto alla computer grafica,
- monitor a fosfori verdi come terminale video e video grafico non a colori;
- televisore a colori per il display di immagini colorate in alta e bassa risoluzione.

Figura 8 - Le figure rappresentano restituzioni prospettiche della casa Domino. Esse sono realizzate mediante plotter piano. L'organizzazione del sistema e dei dati permette di ottenere l'oggetto architettonico smembrato nei suoi componenti principali al fine di uno studio più dettagliato dello stesso.

È in via di acquisizione un plotter piano e la relativa interfaccia, mentre di fatto la potenza del mini è stata portata a 64K con l'introduzione dell'uso del Pascal.
L'ingombro fisico dell'intero sistema è di circa quattro metri quadrati. Non ci sono particolari necessità per quanto riguarda l'alimentazione e la climatizzazione.

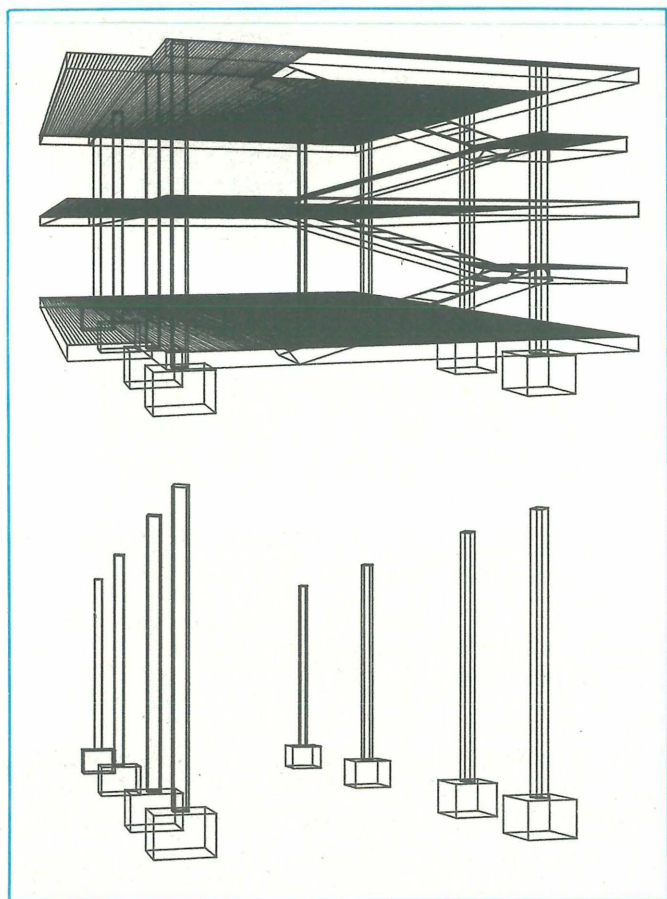
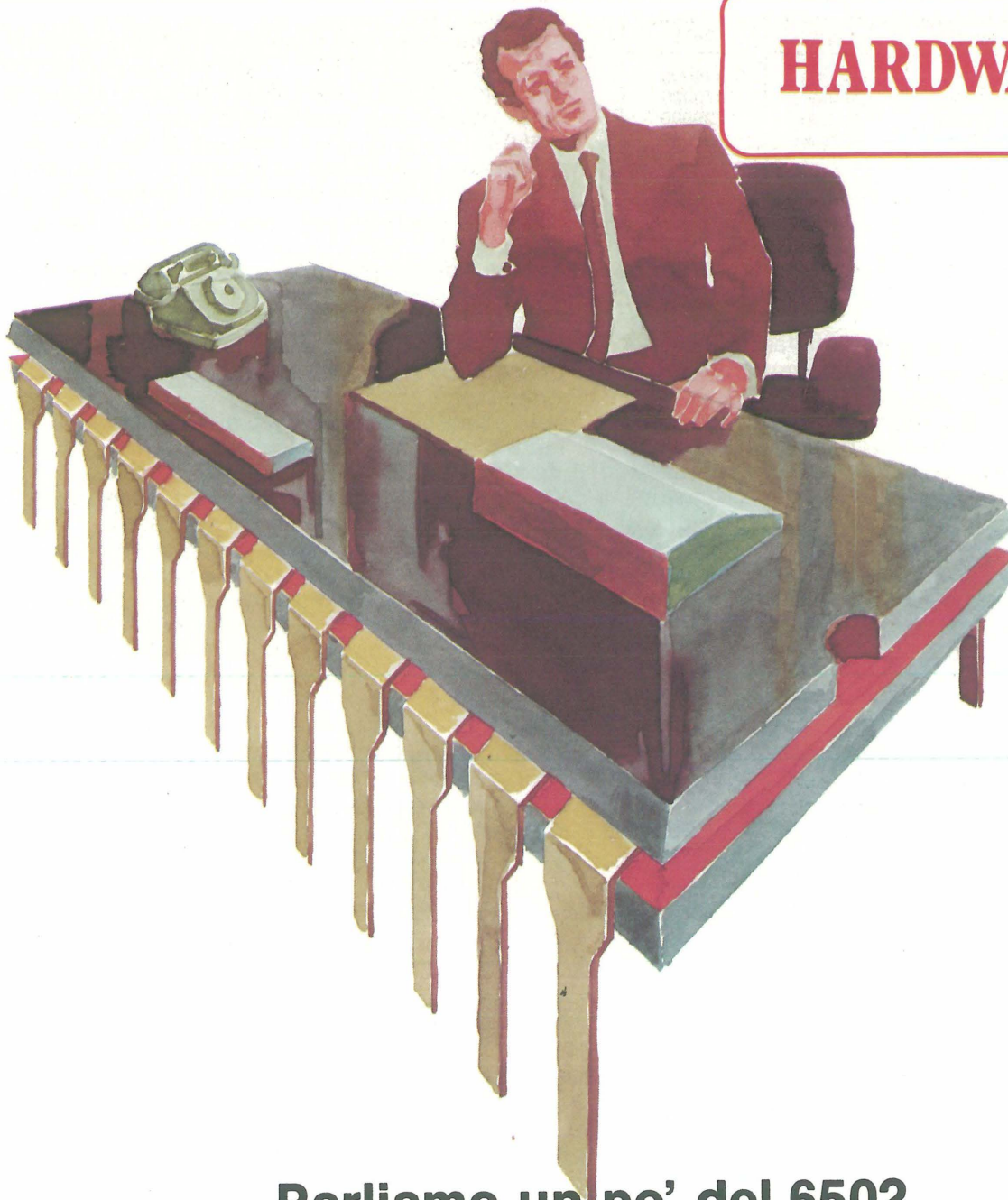
Conclusioni

L'uso della computer grafica nella elaborazione di dati ed in generale nella trasmissione di informazioni è legato al fatto che per ottimizzare l'uso dell'informatica è necessario poter comunicare in maniera semplice, immediata e molto potente quanto dall'informatica viene elaborato.
In altri termini ad un'automazione nell'elaborazione dell'informazione, ad una moltiplicazione delle potenzialità di calcolo, ad un accorciamento dei tempi di calcolo non corrisponde d'altra parte un cambiamento sostanziale dell'utilizzatore al quale le informazioni sono dirette e che è in grado di fruirle essenzialmente mediante l'udito, la visione di figure e la lettura di testi.
È però dimostrato che nulla è più potente, per una comunicazione immediata e comprensibile, dell'immagine. Pertanto oggi si tende a far disegnare i computers. ■

Nota

Tutte le immagini sono state realizzate con il sistema presentato nell'articolo ed illustrato nella Figura 3.
L'autore desidera ringraziare l'amico Harry Loatz che ha aiutato e promosso lo sviluppo del sistema stesso.
Per ulteriori e più dettagliate informazioni rivolgersi all'autore Roma, Via del Boschetto, 29.

La Rockwell inaugura un impianto per prodotti microelettronici

A El Paso (Texas) la Rockwell International ha aperto un nuovo impianto per l'assemblaggio ed il test di sottosistemi microelettronici. L'impianto, altamente automatizzato, occuperà approssimativamente 300 persone, in maggior parte rappresentate da nuove assunzioni.
La Rockwell è stata inoltre incaricata dall'U.S. Air Force di sviluppare un sistema di memorie a bolle magnetiche di elevata densità per applicazioni aerospaziali (4 milioni di bits trasferibili al secondo). Il contratto ha un valore di 1,7 milioni di $. 30 mesi il tempo concordato.

Parliamo un po' del 6502

di **B. Carbone**

Premessa

Certamente non è molto facile, dopo essere stati iniziati con un dato microprocessore, accettare l'idea che ve ne sia qualcun altro più idoneo per le proprie applicazioni, e magari meno costoso. È vero che alla padronanza d'uso di un microprocessore si arriva dopo un certo tempo, vissuto più o meno felicemente, e dopo aver speso non poco in apparecchiature la cui validità è spesso in stretta dipendenza col tipo di microprocessore usato. Nonostante questo tuttavia, correrebbe il rischio di arroccarsi in posizioni sempre più difficilmente sostenibili che, adducendo giustificazioni che alla fine si riducono a quanto si è detto, portasse avanti il discorso della validità del "proprio" microprocessore, magari ignorando alternative piu valide nel proprio campo di applicazione. I discorsi di divulgazione delle caratteristiche di microprocessori "giovani" trovano meno diffidenti coloro

Microprocessore	3° Trimestre 1977	4° Trimestre 1977	1° Trimestre 1978
6502	210000	450000	635000
8080	317000	360000	470000
6800	131000	280000	290000
Z80	50000	75000	160000

Figura 1 - Volumi di produzione di alcuni microprocessori (Stima della Data Quest Inc.).

che col microprocessore vivono i primi momenti; benché privi di esperienza, costoro sono giustamente animati da un desiderio di conoscenza verso prodotti meno gloriosi ma altrettanto affascinanti.
Si vuole in questo articolo mettere a fuoco brevemente le caratteristiche di uno di quei microprocessori che certo non possono fregiarsi dell'etichetta di essere stati "i primi", e che anzi hanno visto la luce in una situazione di mercato abbastanza difficile, caratterizzata dal quasi monopolio di pochissime case costruttrici di circuiti a larga scala di integrazione: il 6502.
Del grosso successo commerciale che ha saputo conquistarsi, e che solo alla luce della considerazioni appena fatte assume il suo significato, vediamone le ragioni.

Il microprocessore 6502

Nato in casa MOS-Techonology, ha trovato presto seconda sorgente nella Synertek e nella Rockwell. Si può dire che ha subito raccolto consensi in molte sedi. Una stima compilata dalla Data Quest Inc., riportata in Figura 1, dà un'idea della quantità delle richieste. Ci si trova di fronte a volumi di produzione che si sono più che triplicati in otto mesi e che, in assoluto, assumono posizioni di indiscutibile preminenza.
Non si può parlare certo a questo livello di maturità del prodotto, ma in ogni caso non stiamo discutendo di un oggetto non sufficientemente collaudato.
Il 6502 in effetti è nato come evoluzione del 6501. Disegnato quest'ultimo per essere pin-compatibile con il 6800, si defferenzia sostanzialmente della CPU Motorola per il numero di istruzioni e i diversi tipi di indirizzamento.
Da quest'ultimo punto di vista, il 6502 si pone in posizione decisamente più avanzata.
Una caratteristica del 6502 di tipo hardware, che finisce per diventare una diversità tra i due, è, per chi conosca il 6800, la funzione collegata al pin 39 (Three-State Control).
Non è possibile infatti avere qui una condizione con stato di alta impedenza sui bus indirizzi e dati contemporaneamente, ma solo sul bus dati.
Ciononostante, con l'uso di opportune tecniche, non sono impedite configurazioni con DMA.
Inoltre il 6502, rispetto al predecessore 6501 e al 6800, è fornito dell'oscillatore e dei drivers di clock, inclusi nello stesso integrato. La base dei tempi è in questo modo fissata all'esterno semplicemente con una rete RC oppure, dovendosi rispettare esigenze di stabilità di frequenza, con un quarzo. In queste condizioni il 6502 può operare a 2 MHz.
Il 6502 è in ogni caso *bus-compatibile* con il 6800. Questo permette al nostro microprocessore di collegarsi con quel gran numero di chips periferici che fanno corona al 6800 e che spesso hanno portato a preferirlo ad altri. Le specifiche di tensione sono infatti le stesse (singola alimentazione a 5 volts) e i de-

Architettura Standard

Ciclo	Bus Indirizzi	Bus Dati	Operazione
1	PC	codice operazione	fetch codice operazione
2	PC + 1		decodifica codice operazione
3	PC + 1	parte bassa indirizzo dato	fetch parte bassa indirizzo dato
4	PC + 2	parte alta indirizzo dato	fetch parte alta indirizzo dato
5	indirizzo completo dato	dato	carica il dato
6	PC + 3		esegui l'operazione
7	PC + 3		deposita il risultato nell'accumulatore
8	PC + 3	codice operazione successiva	fetch codice operazione successiva

Architettura con pipelining

Ciclo	Bus indirizzi	Bus dati	Operazione
1	PC	codice operazione	fetch codice operazione
2	PC + 1	parte bassa indirizzo dato	decodifica codice operazione/fetch parte bassa indirizzo dato
3	PC + 2	parte alta indirizzo dato	fetch parte alta indirizzo dato
4	indirizzo completo dato	dato	carica il dato
5	PC + 3	codice operazione successiva	esegui l'operazione/fetch codice operazione successiva
6	PC + 4	parte bassa indirizzo dato	deposita il risultato nell'accumulatore/decodifica codice di operazione successiva/fetch parte bassa indirizzo dato

Figura 2 - Effetti della struttura *pipeline* nell'esecuzione di un'istruzione con riferimento alla memoria.

vices sono trattati come zone di memoria. Questa caratteristica è estremamente importante quando si rileva la potenza delle istruzioni previste e come esse, in grandissima parte, possono operare indifferentemente con registri interni e zone di memoria. Inoltre anche nel 6502 si può trovare una struttura ad interrupt di tipo mascherabile e non mascherabile (il microprocessore può non "inchiodarsi" mai sul software!).

Esame comparativo del 6502

È necessario, prima di effettuare un esame comparativo del 6502, fare alcune considerazioni. Spesso ci si riferisce, nel confronto tra diversi tipi di microprocessore, a caratteristiche che, a volte, non sono interpretate in modo corretto. A questo scopo non è superfluo precisare che, per quanto riguarda

- *il numero delle istruzioni,*
  si è accertato che sono pochi i codici operativi maggiormente usati nei programmi applicativi e che quindi portano via gran parte del tempo macchina. Sono queste per esempio le operazioni di test, di lettura/scrittura in memoria. Non bisogna quindi dare troppa importanza alla ricchezza di istruzioni che un microprocessore può accettare; molta importanza invece va data a come sono implementate quelle poche ma usatissime istruzioni. Non dimentichiamo inoltre che un linguaggio eccessivamente ricco crea difficoltà di apprendimento e carica il compilatore assembler.

- *il numero dei registri,*
  è invece vero che, maggiore è il numero dei registri interni, più veloce si può prevedere l'esecuzione del programma.
  Notoriamente infatti l'esecuzione di operazioni su registri è più breve di quella su memoria. Ha notevole importanza sotto questo aspetto la possibilità di usare i registri come registri indice. Oltre infatti ad abbreviare i tempi di programmazione, si ottengono in questo caso programmi molto più veloci ed efficienti.

- *la frequenza di clock,*
  essa non ha alcuna importanza ai fini di una corretta valutazione delle prestazioni di un microprocessore. In alcuni microprocessori infatti il clock scandisce cicli di memoria, in altri stati macchina.
  La frequenza di clock quindi assume il giusto significato solo quando si ha presente l'architettura interna del microprocessore. (Un microprocessore con velocità di clock elevatissima può essere più lento se per quell'istruzione ha la necessità di passare per un numero più alto di stati macchina).

Architettura interna

Utilizzando un clock a fase singola a 2 MHz, il 6502 garantisce prestazioni brillanti grazie alla sua architettura interna di tipo *pipeline*. Questa permette di sovrapporre cicli di fetch a quelli di interpretazione dei dati da memoria. Un esempio dimostrativo è presentato in Figura 2. Si può vedere che, per eseguire l'istruzione *ADC* (Addizione con riporto), con una struttura interna tipo *pipeline* si possono risparmiare tre cicli macchina.

Istruzioni

Il numero di istruzioni del 6502 è ridotto a 56, ma tutte sono molto potenti, perché:

1) fanno riferimento diretto alla memoria
2) possono essere associate a indirizzamenti efficienti .

Per quanto riguarda il primo punto, infatti, c'è da dire che, a differenza dei microprocessori più comuni, quasi tutte le istruzioni possono far riferimento alla memoria. Oltre infatti ai soliti *Store/Load*, possono indirizzare la RAM istruzioni di *MOVE, COMPARE,* aritmetiche, logiche e di controllo. (Non siamo costretti quindi ai tediosi carica registro/effettua l'operazione relativa all'indirizzo di memoria caricato).
Il secondo punto merita un discorso più lungo, in quanto le capacità di indirizzamento sono fra le caratteristiche più apprezzate del 6502.
A questo proposito possiamo riferirci alla Figura 3 per fare un rapido confronto con quanto è offerto dai microprocessori più usati.
Vale la pena di fermare in particolare l'attenzione su tre tipi di indirizzamento:

1) Pagina zero
2) Indicizzato
3) Relativo

1) Col primo tipo di indirizzamento al codice operativo che fa riferimento ad una locazione di memoria segue la parte bassa dell'indirizzo di memoria. La parte alta viene forzata dal 6502 a zero. L'intera istruzione si accorcia di un byte e l'esecuzione si completa con un ciclo di clock in meno. Questo è il motivo per cui il 6502 non teme confronti con microprocessori con numerosi registri interni in quanto, se volete, esso ne ha 256 in RAM!
2) È inutile dilungarsi sui vantaggi che la disponibilità di registri indice ha sulla immediatezza di programmazione ed esecuzione dei programmi stessi. In cicli ripetitivi la disponibilità dei due registri indice, come nel 6502, si rileva preziosissima (peccato che siano ad 8 bits!).
   I benefici sono ancora più esaltati dal fatto che l'uso dell'indirizzamento a pagina zero non impedisce lo sfruttamento dei registri stessi come registri indice.
3) L'indirizzamento relativo, associato alle istruzioni di tipo branch (l'80% dei branch fa riferimento alla

Modi di indirizzamento	6502	8080	6800	Z 80
immediato	*	*	*	*
assoluto	*	*	*	*
riferimento a registri	*	*		*
pagina zero	*	limitato (restart)	*	limitato (restart)
pagina zero, indicizzato da X	*			
pagina zero, indicizzato da Y	*			
assoluto, indicizzato da X	*		*	*
assoluto indicizzato da Y	*			*
implicito	*	*	*	*
indicizzato indiretto	*			
indiretto indicizzato	*			
relativo	*		*	*
assoluto indiretto	*	*	*	*

Figura 3 - Modi di indirizzamento in alcuni microprocessori.

stessa pagina), dà luogo anch'esso a programmi più brevi nell'occupazione di memoria e nell'esecuzione.
In particolare, l'esecuzione dei branch sul 6502, per come è stata implementata l'istruzione, è particolarmente veloce (2 μ sec contro i 4,8 μ sec dello Z80).

Stack pointer

Lo stack pointer del 6502 è ad 8 bit e ha caratteristiche di caricamento e di aggiornamento semplificate. Infatti è molto difficile che possa risultare necessario uno stack più alto di 256 bytes. Non avviene qui quello che accade ad esempio per il 6800, cioé il trasferimento automatico di tutti i registri nello stack alla ricezione di un interrupt. Il trasferimento automatico interessa solo il programma counter ed il registro di stato, lasciando al programmatore la facoltà di continuare o no nell'operazione di salvataggio.

Prove al banco

Se si vogliono evitare discorsi di architettura interna, numero d'istruzioni, indirizzamenti etc, e si vuole avere tuttavia un'idea delle prestazioni di un microprocessore, niente è più convincente e facilmente recepibile dell'esame dei risultati delle cosiddette "prove al banco". Si tratta di far eseguire a vari tipi di microprocessori più programmati test che riproducano le condizioni più frequentemente riscontrabili in programmi applicativi. I parametri che qualificano il microprocessore sono:

- la zona di memoria occupata
- la velocità di elaborazione.

Si riportano in Figura 4 alcuni dati di confronto tra 6502, 8080, 6800, e Z80. Si sono usati per ciascun microprocessore clock con frequenze riportate in riquadro. I tests sottoposti (abbastanza pochi per la verità per una valutazione globale, ma tuttavia indicativi) sono stati:

- *Ricerca di carattere*
  Si trattava di trovare in un vettore di 40 bytes posto in una zona qualsiasi della memoria un elemento uguale ad un altro situato anch'esso in una zona di memoria.
- *Shift a destra*
  Si trattava in questo caso di una parola a 16 bits e di uno shift di 5 posizioni.
- *Trasferisci e modifica una zona di memoria*
  Era interessato al trasferimento da una zona di memoria ad un'altra un blocco di 40 parole a 16 bits (2 bytes) dopo essere state ciascuna ruotata a destra di 2 bits.

Le prestazioni, quanto ad occupazione di memoria e velocità di trasferimento sono state in figura tutte normalizzate a quelle del 6502. Ciò significa che valori inferiori ad uno qualificano il prodotto interessato a livelli più elevati del 6502; il contrario per valori maggiori di uno.
Dati indicativi un po' più completi si possono ancora trarre da prove effettuate da *Kilobaud*, una rivista americana sufficientemente obbiettiva da farci credere che in effettive prove di applicazione il 6502 si comporta in modo così brillante come non ci si aspetterebbe da un esame molto superficiale delle sue caratteristiche. Ci si riferisce in questo caso a prove di velocità di interpreti BASIC su più tipi di microsistemi in commercio, effettuate con 7 programmi test (Figura 5). Si può vedere come microcalcolatori che fanno uso di CPU 6502 abbiano dato luogo a risultati davvero esaltanti, in assoluto addirittura superiori dello Z80 a 4 MHz.

Costo

Si è voluto riservare per ultimo l'esame di questa caratteristica perché il costo di un dispositivo è uno degli elementi più importanti per decidere dell'uso di un certo tipo di microprocessore. Non si vuol parlare evidentemente della persona che, dovendo acquistare un intero sistema, si lascia assalire da dubbi sul prezzo della CPU. In queste condizioni infatti è talmente grande la sproporzione tra costo dell'intero sistema e costo della sola CPU, che il discorso della economicità della CPU non ha senso. Si vuole invece parlare di quell'ampia fascia di utilizzatori che animano il mercato delle applicazioni spicciole, quel mercato cioé in cui le applicazioni sono così centralizzate nella CPU che differenze di costo di quest'ultima anche di poche lire, essendo molto grande il volume di produzione, possono risultare determinanti nella scelta di un certo tipo di microprocessore (video-games, apparecchiature per comunicazioni, di strumentazione, etc.).
Le caratteristiche delle quali il 6502 si giova per presentarsi come componente altamente competitiva anche in termini di costo sono sostanzialmente:

- La più piccola superficie del chip
  Questo evidentemente porta alla produzione di un maggior numero d'integrati a parità di dimensioni del wafer di silicio. Per poter acquisire questa caratteristica sono stati fatti in fase di progettazione parecchi sforzi. Si è trattato di trovare il miglior compromesso tra set di istruzioni, architettura, di-

Programma test	Occupazione di memoria				Tempo di esecuzione				Frequenza di clock			
	6502	8080	6800	Z80	6502	8080	6800	Z80	6502	8080	6800	Z80
Ricerca carattere	1,00	1,82	1,64	1,45	1,00	2,28	1,82	0,99	2 MHz	3 MHz	2 MHz	4 MHz
Shift a destra	1,00	1,27	1,33	0,93	1,00	2,60	1,17	0,91	2 MHz	3 MHz	2 MHz	4 MHz
Trasferisci e modifica un blocco	1,00	1,96	1,56	1,03	1,00	3,18	2,06	1,32	2 MHz	3 MHz	2 MHz	4 MHz

Figura 4 - Risultati di "prove al banco".

	CPU Numero di benchmark								
	Tipo	1	2	3	4	5	6	7	Note
1. OSI 8K BASIC (ver 1.0. rev 3.3): OSI Challenger @ 2 MHz	6502	9	4.6	8.2	9.3	10.0	14.8	21.6	b
2. Zapple 8K BASIC (1.1): Altair 8800 a, Cromemco Z-80 4 MHz, 1 wait state	Z-80	9	5.9	13.0	13.5	14.8	22.7	32.7	a
3. OSI 8K BASIC (ver 1.0, rev 3.2); OSI Challenger @ 1 MHz	6502	1.4	8.6	15.9	17.8	19.3	28.7	42.2	a
4. OSI 8K BASIC (ver 1.0, rev 3.3); OSI Challenger @1 MHz	6502	1.6	8.9	16.2	18.2	19.7	29.2	42.9	b
5. PET BASIC; Commodore PET 2001 (prototipo)	6502	1.7	9.8	18.6	20.4	22.1	32.6	51.3	b
6. Zapple 8K BASIC (1.1): Altair 8800a, TDL ZPU @ 2 MHz	6502	1.7	9.5	20.6	21.7	23.7	36.2	51.8	a
7. Altair 8K BASIC (4.0); Altair 8800b	8080	1.7	10.2	21.0	22.5	24.3	36.7	52.4	a
8. Altair 8K BASIC (3.2); Altair 8800a	8080	1.7	10.3	21.4	23.1	24.8	37.3	52.8	a
9. Altair 8K BASIC (3.0); Imsai I-8080	8080	1.6	10.6	22.0	23.7	25.4	38.3	57.1	a
10. Digital Group Z-80 Maxi-BASIC (1.0); Digital Group Z-80 2.5 MHz	Z-80	1.8	7.5	21.2	25.1	26.9	40.3	58.5	a
11. Altair 12K Extended BASIC (4.0); Altair 8800b	8080	1.9	7.5	20.6	20.9	22.1	37.0	58.5	a
12. Altair Disk Extended BASIC (4.0); Altair 8800b	8080	1.9	7.5	20.6	20.9	22.1	36.9	58.5	a
13. North Star BASIC-FPB (ver 6); Altair 8800a, North Star Fl. Pt. Board	8080	1.9	9.1	18.4	18.5	20,9	36.1	59.4	b
14. Altair 12K Extended BASIC (3.2); Altair 8800a	8080	1.9	8.9	21.8	23.0	24.8	39.3	60.7	a
15. Altair Disk Extended BASIC (3.4); Altair 8800b	8080	1.9	8.8	21.7	22.8	24.7	39.6	61.6	a
16. Altair 4K BASIC (4.0); Altair 8800b	8080	1.9	15.1	26.0	28.9	31.7	44.5	62.1	a
17. Compal-80 10K BASIC; Compal-80	8080	2.0	9.3	23.4	24.6	26.3	42.1	65.7	a
18. Processor Tech BASIC 5; Digital Group 8080	8080	3.6	10.5	27.5	30.9	33.2	51.3	67.4	a
19. Compucolor 8K BASIC; Compucolor 8001	8080	2.1	13.1	27.0	29.0	31.3	47.5	67.8	a
20. Digital Group 8080 Maxi-BASIC (1.0); Digital Group 8080	8080	2.2	9.2	26.4	31.2	33.5	49.9	72.3	a
21. North Star BASIC (ver 6); Altair 8800a	8080	2.3	9.5	26.6	31.3	33.7	50.6	73.8	b
22. Poly 11K BASIC (ver 9V27); Poly-88 System 16	8080	2.5	10.2	29.0	34.0	36.5	54.0	79.0	b
23. Altair 680 8K BASIC (3.2); Altair 680b	6800	2.5	16.3	30.7	33.4	36.3	55.9	81.8	a
24. Poly 11K BASIC (ver A00); Poly-88 System 16	8080	2.5	11.3	31.3	36.1	39.3	58.6	87.6	b
25. Tektronix Level 5 BASIC; Tektronix 4051	6800	4.8	14.0	33.0	36.2	40.7	68.8	103.8	b
26. Micropolis BASIC (1.1); Altair 8800a, Mod II Micropolis disk	8080	8.7	19.9	50.4	54.1	58.2	109.9	146.4	a
27. IBM 5100 BASIC; IBM 5100	??	4.0	20.5	56.5	54.3	58.1	87.0	172.8	a
28. Southwest Tech 8K BASIC (1.0); SWTPC 6800	6800	14.9	24.7	96.1	105.3	109.8	174.1	204.5	a
29. Imsai 8K BASIC (1.31); Imsai I-8080	8080	7.5	28.2	66.4	78.5	88.1	140.1	235.6	a,c
30. Imsai 8K BASIC (1.3); Imsai I-8080, 1702 ROM, 1 1/2 wait states	8080	11.5	39	92	110	121.5	191	320	b
31. SCELBAL BASIC; MIKE-2	8008	312	369	460	515	760	1814	2151	b

Nota a: Temporizzazione fatta dagli autori.
Nota b: Temporizzazione fornita da altri.
Nota c: Questa è una versione di 1,3 che lavora in RAM. Non ancora rilasciata dalla Imsai.

Figura 5 - Tempi di benchmark con interpreti BASIC (da *Kilobaud Magazine*).

segno logico e lay-out del chip. Alcuni tra i vari accorgimenti usati sono:

1) Il ridotto set di istruzioni, che permette alla parte ROM di decodifica del chip il raggiungimento di dimensioni molto ridotte.
2) La dislocazione dei registri, che sono stati impacchettati accanto alla ROM
3) Il disegno delle linee di bus, che girano intorno ai registri piuttosto che occupare zone non usate all'intero chip.

- La più bassa dissipazione del chip.
Questo permette l'uso di economici contenitori plastici in luogo dei ceramici, e garantisce una maggiore affidabilità.

- L'uso di una singola alimentazione.
Sono evitati i costi per alimentatori multitensione.

- L'integrazione dell'oscillatore nello stesso chip CPU, che elimina l'impiego di un circuito logico di temporizzazione esterna.

- La disponibilità di contenitori plastici a 28 pins, con riduzione di costo della stessa CPU e minore occupazione di area del circuito stampato.
Di questi contenitori a numero di pins ridotto sono munite le CPU della famiglia 650X di tipo 6503, 6504, 6505, 6506 e 6507. Esse si differenziano per possibilità di indirizzamento e utilizzazione diversa, dal punto di vista funzionale, dei pins disponibili. Risultano usatissimi nelle realizzazioni più economiche.

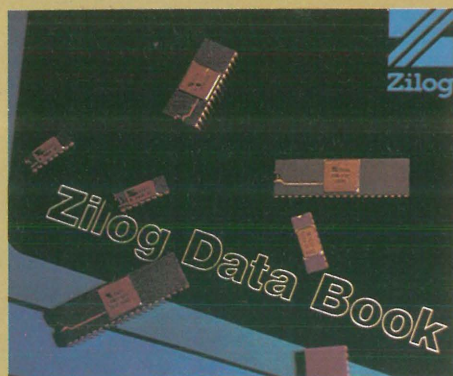
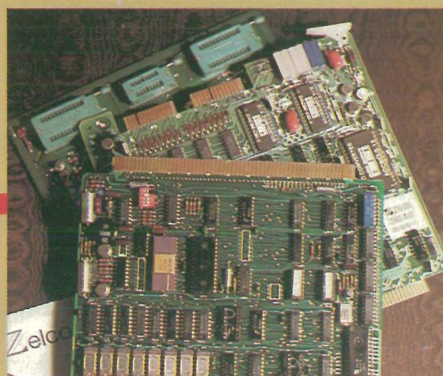
Utilizzatori

Molti sono gli utilizzatori OEM che hanno disegnato microsistemi a singola o più schede intorno al 6502. A parte la MOS-Technology con il famosissimo KIM-1, la Rockwell con l'AIM-65 e la Synertek con la popolare SYM1 (quest'ultima offerta anche in versione senza tastiera e visualizzatore SM100 o in versione ridotta CP110), che producono direttamente il 6502, basta ricordare tra i più noti:

- La Compas Microsystems, con i suoi CSB1 e CSB2
- La Ohio Scientific, con il suo Challenger dal BASIC velocissimo
- La Apple Computer, con il suo Apple II, microelaboratore a singola scheda con generatore video e possibilità di 48 Kbytes di RAM e 12 Kbytes di ROM
- La Commodore, con il suo diffusissimo PET.

L'elenco é evidentemente incompleto.
In ogni caso é da pensare, dato il crescente numero di costruttori più o meno grandi che hanno deciso di legare le fortune dei loro prodotti al 6502, che questo microprocessore troverà sempre più diffusa applicazione, soprattutto nelle fasce basse di mercato (ci si riferisce al costo per unità di prodotto), non ultimi, negli home computers, date le sue caratteristiche di semplicità, efficienza e costo ridotto. ■

E' DURO PER ALCUNI DOVER AMMETTERE DI AVER PERSO IL "QUASI MONOPOLIO". I FATTI SONO QUELLI CHE CONTANO:

LA REALTÁ DEI MICRO COMINCIA DALLA " "

DUE FISIONOMIE DIVERSE MA UNICHE SOTTO MOLTI ASPETTI

Zilog

- E' l'unica casa completamente dedicata ai micro.
- Progetta tenendo presente i problemi globali ed in funzione di questi ottimizza i particolari perciò:
  - vi offre di meglio come componenti, come piastre, come sistemi e come software.
- Ha alle spalle il più forte gruppo finanziario del mondo;
  - ciò le consente di avere i migliori specialisti mondiali di hardware, di software e di sistemistica e da
  - la sicurezza di disporre dei finanziamenti necessari a conservare la supremazia nel proprio campo (per questo Vi sono ditte anche in italia, che già nel 1979 hanno impostato la produzione di centinaia di migliaia di macchine con lo Z80).

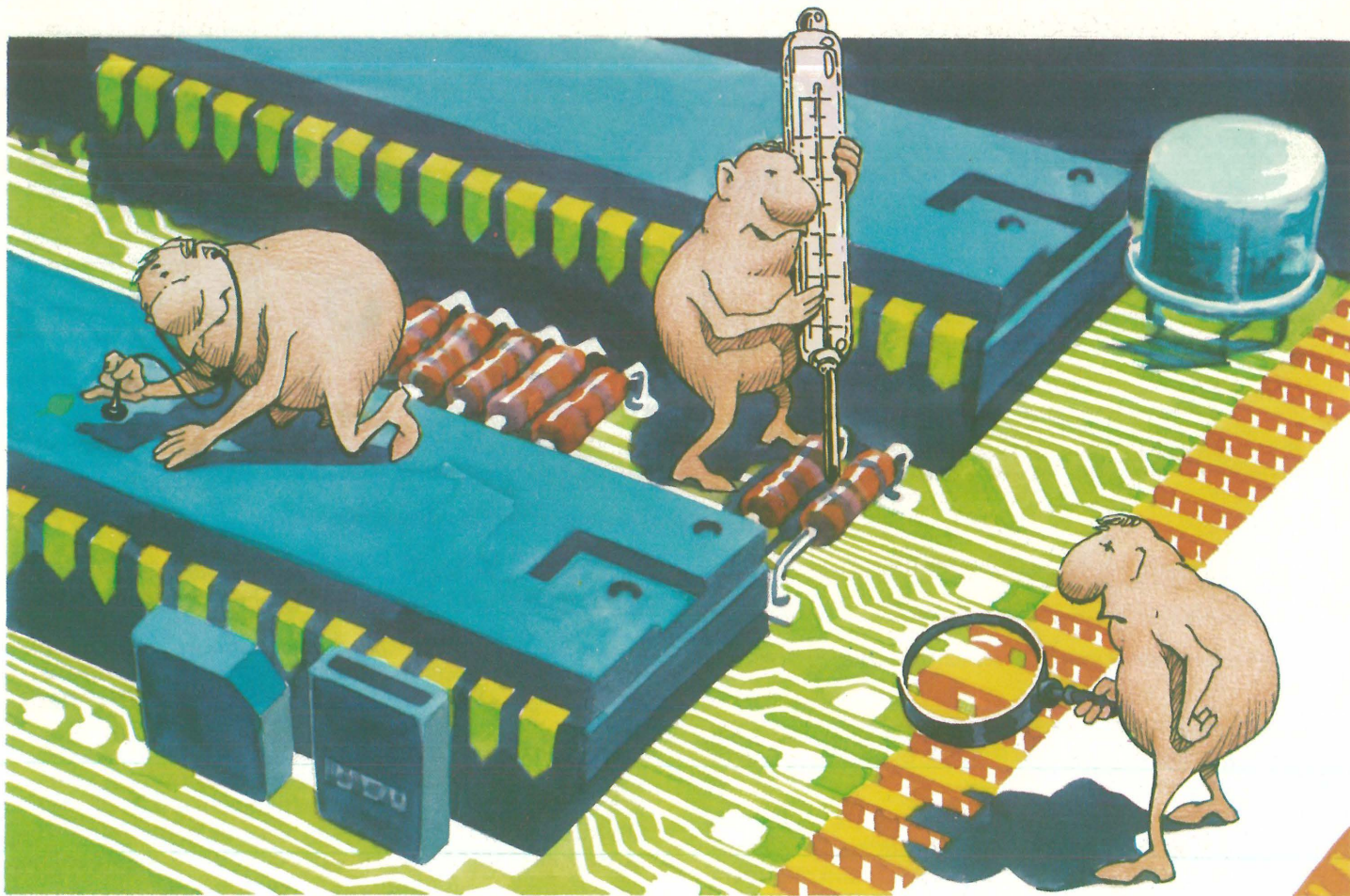
- E' esclusivamente dedicata ai micro e dà un supporto che va dalla fornitura di componenti, agli accessori (terminali, stampanti etc.) per sistemi
- E' composta da specialisti e non da venditori per offrire un servizio più rapido e non una assistenza che si ferma di fronte ai problemi nuovi;
- Usa essa stessa intensamente i calcolatori che vende, li conosce a fondo e produce software sofisticato per se e per gli altri.
- Ha una organizzazione agile che le permette di non ricaricare spese eccessive sui prodotti che vende e di rispondere in "real time". (Provare per credere).
- Ha una consegna pronta; ciò significa che non dovete aspettare due mesi per un SIO o una CPU e neppure per un calcolatore.
- Vi assiste direttamente per riparazioni o consulenza qualificata.

La nostra pubblicità maggiore è fatta da tre elementi:

1 - I nostri clienti **2 - La nostra qualità** **3 - I nostri prezzi**

Fate i confronti: per noi basta telefonare al 02/803336/804247

(nelle zone di Roma al 06/8102836)

Zelco S.r.l. Via V. Monti 21, 20123 Milano

Il Picocomputer: scheda di unità centrale

Parte III. di **D. Del Corso**

Questo articolo conclude la descrizione della scheda di unità centrale per Picocomputer. Viene riportato anche un breve programma di monitor per usare l'insieme finora descritto (scheda-picoperiferico) come piccolo sistema autonomo.
La serie continuerà nei prossimi numeri con la presentazione di interfacce per registratore e TV, di schede di espansione (RAM, ROM, I/O), e con la pubblicazione di una serie di esercizi studiati per familiarizzare l'utente con il picomputer.

Montaggio e collaudo

Lo schema di pag. 46-47 (BIT n° 4) può essere montato anche solo in parte, per esempio senza buffer, o senza memoria ed I/O locale.
Per la versione minima (Z 80 con interfaccia MU-BUS) bastano gli integrati n° 10, 13, 14, 15 e 18. I buffers n° 1, 2, 3, 4, 5, e 6 possono essere omessi se la scheda é usata da sola, o sostituiti con ponticelli per piccoli sistemi (fino a 3 schede complessive). Gli integrati sono indicati con i simboli funzionali (1), che permettono di specificare la operazione logica implementata da ciascuna porta.
A montaggio terminato é possibile eseguire alcune semplici operazioni di collaudo statico, che non richiedono l'uso di strumentazione sofisticata. Conviene per questo realizzare, anche solo su un breadboard, il circuito di Figura 1. Con questo semplice "pannello di controllo", i comandi STOP e STEP

GAME TORINO

a cura del Gruppo Amatoriale Micro-Elaboratori, Torino *e del Laboratorio Didattico di Elettronica Applicata dell'Istituto di Elettronica e Telecomunicazioni del Politecnico di Torino.*

(1) Questa simbologia è ripresa da *"Sistemi a microprocessore"*, collana Quaderni di Elettronica (Boringhieri).

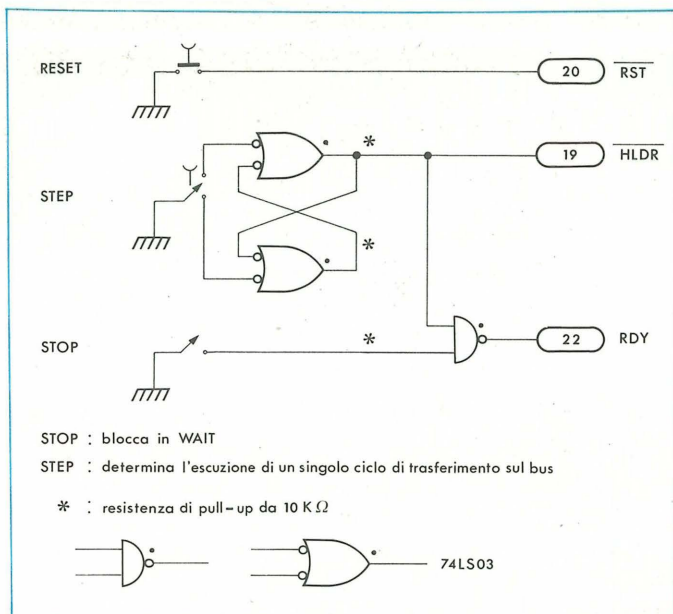



Figura 1 - Esecuzione passo-passo: schema del circuito di controllo.

permettono l'esecuzione passo passo di qualsiasi programma, ed il pulsante RESET invia il segnale di inizializzazione RST.

Il funzionamento a ciclo singolo é basato sull'alternanza di stati di HOLD e WAIT; il microprocessore viene bloccato su ogni ciclo di accesso alla memoria o all'I/O dalla richiesta di WAIT (linea RDY = 0). Premendo il pulsante STEP si porta RDY a 1, e si alza una richiesta di HOLD (linea $\overline{HLDR}$ = 0); il ciclo di trasferimento così si conclude ed il processore va in HOLD. Rilasciando STEP la richiesta di HOLD viene rimossa e si riporta RDY a 0. Si inizia così, e si blocca immediatamente, il ciclo successivo della CPU. La sequenza temporale dei comandi é riportata nella Figura 2. Questo circuito usa linee standard di Bus ($\overline{HLDR}$ e RDY) e funziona con qualsiasi microprocessore con funzioni HOLD e WAIT statiche.
Possiamo ora far eseguire allo Z 80 le varie operazioni (accessi a memoria ed I/O, salti, interruzioni ecc), una alla volta, e verificare con semplici strumenti statici i diversi segnali, sia della CPU, sia sul bus esterno MUBUS (Figura 3). Per visualizzare lo stato delle linee bastano dei LED, usando l'accortezza di interporre delle porte (ad esempio 74LS05) con funzione di buffer, per non caricare eccessivamente i punti in prova.

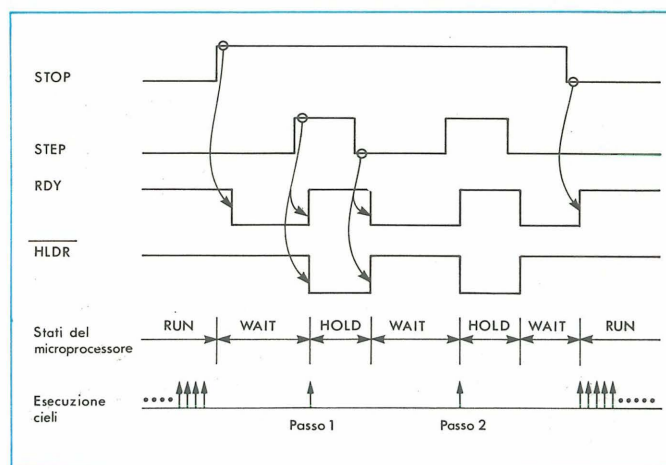



Figura 2 - Esecuzione passo-passo: temporizzazione.

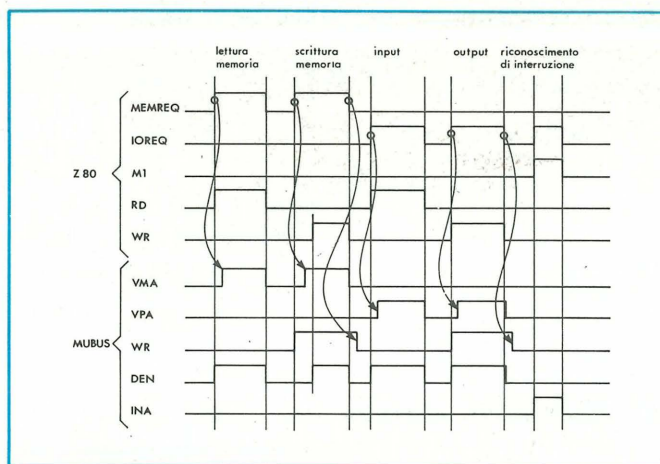



Figura 3 - Codifica Z 80-MUBUS: cicli di trasferimento dati.

E' anche possibile, e utile, realizzare un "pannello di monitoraggio" completo, che presenti contemporaneamente lo stato di tutte le linee di bus: indirizzi, dati e controlli.
Subito dopo aver collegato il circuito di passo-passo, é già possibile far eseguire al picocomputer un programma, e precisamente una sequenza di NOP (No Operation). Infatti, dato che il codice operativo di questa istruzione é 00, basta collegare a massa le linee DATA 0-7 (conviene fare dei ponticelli sullo zoccolo della ROM, e disinserire RAM e buffer dati per evitare conflitti). L'istruzione NOP determina solo un incremento del program counter; tenendo bloccato il microprocessore con il comando STOP, dopo un RESET tutte le linee di indirizzo vanno a 0 (Program Counter = 0000. Con un primo STEP avremo ADD 0 = 1 (PC = 0001), al successivo ADD 0 = 0 e ADD 1 = 1 (PC = 0002) e così via per 65.535 passi (fino a PC = $FFFF_H$).
Nell'esecuzione passo-passo i cicli di rinfresco vengono saltati perché in essi lo Z 80 non testa la linea WAITREQ (RDY nel nostro standard). Togliendo il comando STOP l'esecuzione procede alla cadenza normale ed é possibile osservare i vari segnali su un oscilloscopio.
Questo metodo di collaudo a cicli singoli può essere esteso ad altre istruzioni; é sufficiente forzare sulle linee dati (sempre attraverso i piedini della ROM) il codice operativo della istruzione che si vuol fare ese-

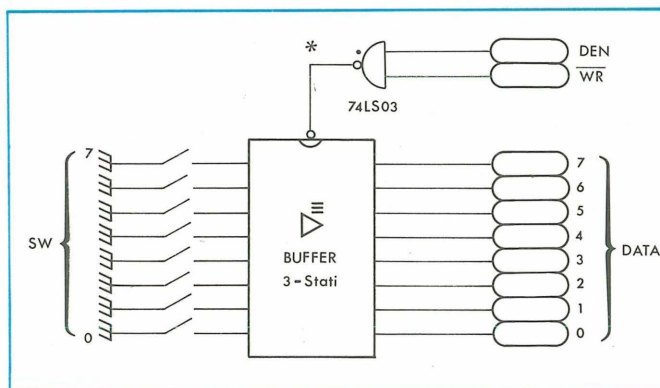
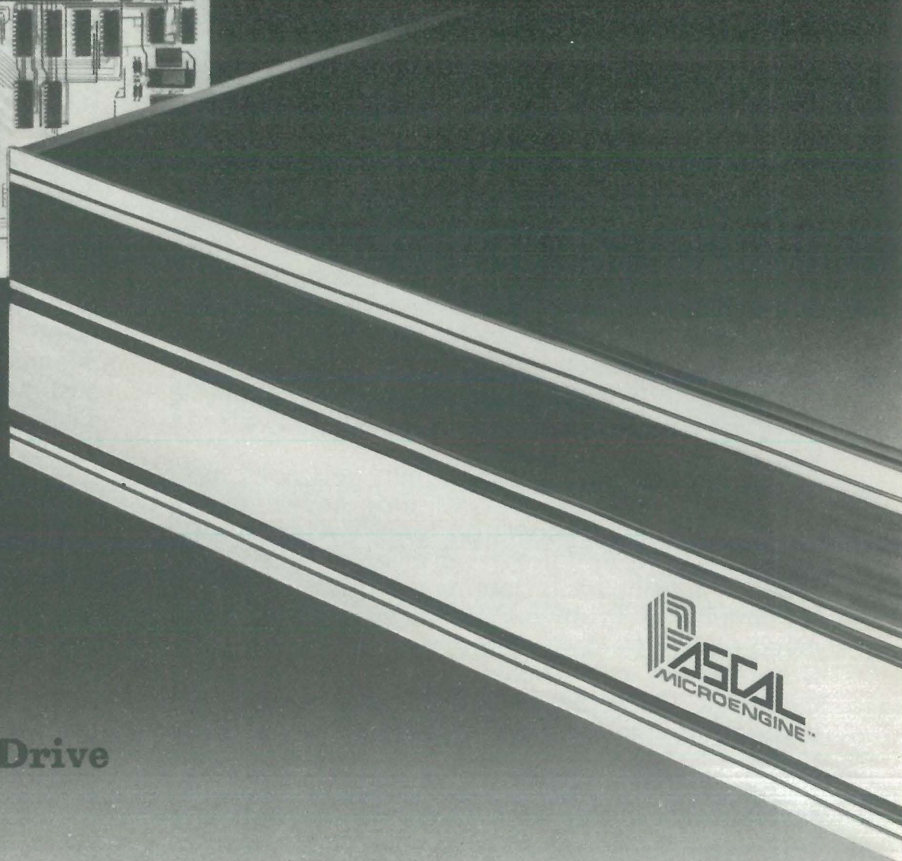
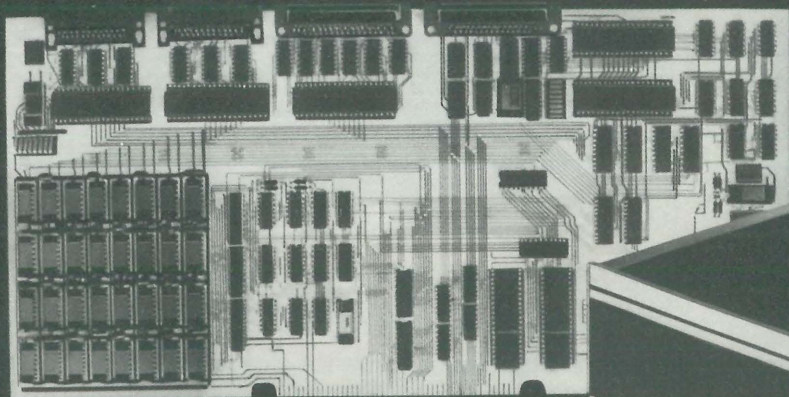



Figura 4 - Circuito per il forzaggio di istruzioni sulle linee dati.

WD/90 Pascal MICROENGINE™

- 64 K Byte RAM
- Floppy Controller per 4 Drive
  singola/doppia faccia
  singola/doppia densità
- 2 Interfacce seriali
- 2 Interfacce parallele
- Usa lo stesso "Chip-set" dell' LSI 11*
- Microprogrammato per eseguire direttamente istruzioni in PASCAL
- Prezzo 10 up:
  sistema completo di software: 2.950.000 Lit. più IVA
  scheda logica: 1.950.000 Lit. più IVA

*LSI è un marchio registrato DEC

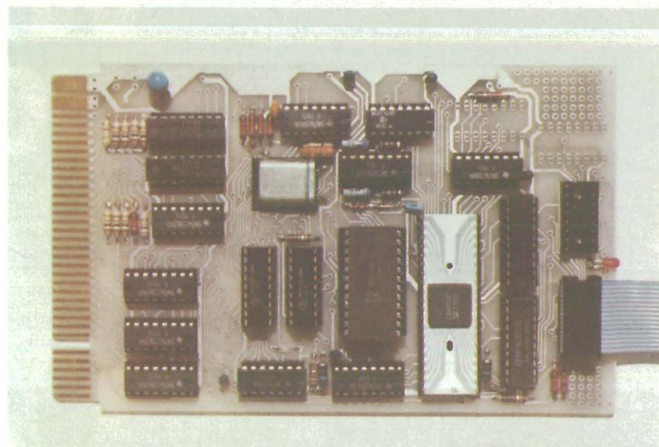
COMPREL s.r.l.

20092 CINISELLO B. (MI) - VIALE ROMAGNA, 1
☎ (02) 6120641/2/3/4/5 - Telex: 332484 COMPRL I

Uffici regionali:
40137 BOLOGNA - P.za Azzarita, 6 - Tel. (051) 551306
50127 FIRENZE - Via T. Mabellini, 4 - Tel. (055) 412018
16033 LAVAGNA (GE) - P.za Marini, 20/10 - Tel. (0185) 301100
60025 LORETO (AN) - Via Dante Alighieri, 26/B - Tel. (071) 977693
35100 PADOVA - Via R. De Visiani, 17 - Tel. (049) 750741
00141 ROMA - Via Muzio Clementi, 58/5 - Tel. (06) 3603463 - 36009
10144 TORINO - Via G. Fagnano, 10 - Tel. (011) 472789

guire. Conviene collegare un banco di 8 interruttori ove predisporre, un byte alla volta, il codice dell'istruzione, dare uno STEP, predisporre il byte successivo e così via.
In questo caso però le istruzioni che comportano cicli di scrittura o di output determinerebbero un conflitto di accesso sulle linee dati. E' quindi necessario interporre, tra bus ed interruttori, un buffer a 3 stati, come in Figura 4. Il circuito di controllo abilita le uscite solo durante le operazioni di lettura (DEN = 1 e WR = 1).
A questo punto il picocomputer può, con molta, molta pazienza, eseguire qualsiasi programma, servire interruzioni, fare in sostanza tutto, ovviamente un passo alla volta. Soffermarsi ad inventare e verificare esercizi e semplici programmi a questo livello é estremamente istruttivo e stimolante, ma anche scomodo e tutt'altro che rapido.
Conviene già dall'inizio avere a disposizione strumenti che permettano di preparare e far eseguire programmi in modo più comodo e veloce. Utensile basilare per questo obiettivo é un complesso hardware (periferici)/software (programmi di debug, assemblatori, ecc.) tale da rendere più semplici ed efficaci, e quindi meno soggetti ad errori, la preparazione ed il collaudo dei programmi di utente.
Questa generica definizione copre praticamente tutti gli "aiuti allo sviluppo", dai sistemi molto complessi che possono lavorare in linguaggi evoluti, fino al limite inferiore costituito dai microcalcolatori più semplici, con periferici di basso costo (ad esempio tastiera e visualizzatore), supportati da programmi (detti *monitor*) che permettono solo di preparare e verificare i programmi di utente direttamente in linguaggio macchina.
Il picocomputer, almeno nella versione presentata in questi articoli, appartiene chiaramente a quest'ultima categoria.

PICOMON .0

PICOMON .0 é il primo, e più semplice, livello di monitor per il picocomputer. Il programma occupa 256 celle di ROM e circa 40 celle di RAM, per lo stack e buffer temporanei, ed utilizza come organo di colloquio con l'operatore una tastiera da 24 tasti ed un display esadecimale ad 8 cifre, gestiti con la tecnica descritta ne "Il picoperiferico" (BIT n° 3).
I collegamenti necessari tra picoperiferico e scheda CPU sono indicati nella Figura 5. Da notare l'aggiunta del pulsante di RESET, non indicato negli schemi di BIT n° 3. Il tasto di seconda funzione non é utilizzato in PICOMON .0.
Questo programma di monitor é organizzato in modo da rendere disponibili anche alcuni sottoprogrammi base (ritardo, scansione, codifica ecc), utilizzabili nei programmi di utente. E' anche possibile ridefinire completamente i comandi, mantenendo la gestione del display e della tastiera. Queste caratteristiche saranno sfruttate negli esempi di applicazione descritti nei prossimi articoli.

Zoccolo su scheda CPU (A)	Funzione	Picoperiferico[1]
1	OUT 4	SCAN 0
2	OUT 5	SCAN 1
3	OUT 6	SCAN 2
4	OUT 7	SCAN 3[2]
5	OUT 0	CAR 0
6	OUT 1	CAR 1
7	OUT 2	CAR 2
8	GND	GND
9	OUT 3	CAR 3
10	IN 1	COD 1
11	IN 0	COD 0
12	RST	– [3]
13	IN 5	– [4]
14	IN 7	STB
15	IN 6	– [4]
16	+ 5V	

[1] V. BIT n°3, Figura 12, pag. 37.
[2] Va collegato a massa nel picoperiferico (display di 8 cifre).
[3] Collegare ad un pulsante non indicato nello schema.
[4] Linee non usate da PICOMON.0

Figura 5 - Collegamenti del picoperiferico.

```
                        PICOMON0        30_OTT_79           PAGE   1
LOC   OBJ CODE M STMT SOURCE STATEMENT                        ASM 5.8

                  1  ;
                  2  ;
                  3  ;
                  4  ;
                  5  ;       MONITOR PER CONSOLLE A TASTIERA
                  6  ;
                  7  ;
                  8  ;
                  9  ;       DEFINIZIONE PARAMETRI
                 10  ;
                 11  ;
                 12  COMTAB  EQU     0BF8H   ;IND. INIZ. TABELLA COMANDI
                 13  SAVSP   EQU     0BE0H   ;SALVA SP
                 14  SAVDE   EQU     0BE6H   ;SALVA DE
                 15  SAVREG  EQU     0BF0H   ;AREA SALVATAGGIO REGISTRI
                 16  INTSER  EQU     0BF5H   ;ROUTINE DI INT SU RST 7
                 17  BUFF    EQU     0BF0H   ;BUFFER DEL DISPLAY
                 18  UTSP    EQU     0BD0H   ;AREA STACK DI UTENTE
                 19  MOSP    EQU     0BE0H   ;AREA STACK DI MONITOR
                 20  BRKSAV  EQU     0BFAH   ;SALVA CELLA DI BREAK
                 21  MAXBUF  EQU     0F4H    ;LIMITE BUFFER DISPLAY
                 22  COMASK  EQU     60H     ;MASCHERA CONTATORE SCANSIONE PARI
                 23  LOMASK  EQU     0FH     ;MASCHERA NIBBLE BASSO
                 24  HIMASK  EQU     0F0H    ;MASCHERA NIBBLE ALTO
                 25  CONADD  EQU     0C00H   ;INDIRIZZO CONSOLLE
                 26  SCMASK  EQU     70H     ;MASCHERA CONTATORE SCANSIONE
                 27  INMASK  EQU     83H     ;MASCHERA BIT STB E CODIFICA COLONNA TASTIERA
                 28  CDMASK  EQU     7       ;MASCHERA COMANDI
                 29  BRKRST  EQU     0DFH    ;RESTART DA BREAKPOINT
                 30  DELY    EQU     80H     ;VALORE DEL RITARDO
                 31  STB     EQU     7       ;FLAG DI TASTO PREMUTO
                 32  ;
                 33  ;
                 34  ;
                 35  ;       INIZIALIZZAZIONI
                 36  ;
                 37  ;
0000  219B00   R 38  INIZ1   LD HL, COMSTA   ;INDIRIZZO INIZIALE TABELLA COMANDI
0003  22F80B     39          LD (COMTAB), HL
0006  21D00B     40          LD HL,UTSP      ;INIZIALIZZA STACK DI UTENTE
0009  22E00B     41          LD (SAVSP),HL
000C  31E00B     42  INIZ2   LD SP,MOSP      ;INIZIALIZZA STACK POINTER DEL MONITOR
000F  00         43          NOP             ;SPAZIO PER ALTRE INIZIALIZZAZIONI
0010  00         44          NOP
0011  00         45          NOP
                 46  ;
                 47  ;
                 48  ;
                 49  ;       PROGRAMMA PRINCIPALE
                 50  ;
                 51  ;
0012  CD3B00   R 52  MAIN    CALL COP        ;CHIAMA  ROUTINE DI SCANSIONE E CODIFICA
0015  18FB       53          JR MAIN
0017  00         54          NOP
                 55  ;
                 56  ;
                 57  ;
                 58  ;       ESECUZIONE DI BREAKPOINT
                 59  ;       SALVA I REGISTRI DEL BANCO PRINCIPALE IN RAM
                 60  ;       PORTA IN DE L'INDIRIZZO DI BREAK
                 61  ;       RIPRISTINA IL CONTENUTO DELLA CELLA DI BREAK
                 62  ;
0018  ED53E60B   63  BREKEX  LD (SAVDE),DE   ;SALVA DE
001C  D1         64          POP DE          ;RICOSTRUISCE L'INDIRIZZO DEL BREAK
001D  1B         65          DEC DE
001E  ED73E00B   66          LD (SAVSP),SP   ;SALVA SP
0022  3AFA0B     67          LD A,(BRKSAV)   ;RIPRISTINA IL CONTENUTO DELLA CELLA
0025  12         68          LD (DE),A       ;DI BREAK
0026  31F00B     69          LD SP,SAVREG    ;INIZIALIZZA SP PER SALVATAGGIO REGISTRI
0029  D5         70          PUSH DE         ;SALVATAGGIO REGISTRI IN SAVREG
002A  F5         71          PUSH AF
002B  C5         72          PUSH BC
002C  3B         73          DEC SP          ;SALTA SAVDE
002D  3B         74          DEC SP
002E  E5         75          PUSH HL
002F  DDE5       76          PUSH IX
0031  FDE5       77          PUSH IY
0033  18D7       78          JR INIZ2        ; AL MONITOR CON IND. DI BREAK IN DE
0035  00         79          NOP
0036  00         80          NOP
0037  00         81          NOP
                 82  ;
                 83  ;
                 84  ;
                 85  ;       SERVE INTERRUPT SU RST 7
                 86  ;
0038  C3F50B     87  INTR    JP INTSER       ;SALTA IN RAM
                 88  ;
                 89  ;
                 90  ;
                 91  ;       AGGIORNAMENTO DEL BUFFER DISPLAY
                 92  ;       COPIA:  D IN BUFF+3,
                 93  ;               E IN BUFF+2
                 94  ;               (DE) IN BUFF+1
                 95  ;
                 96  ;
003B  21F30B     97  COP     LD HL, BUFF+3
003E  72         98          LD (HL),D
003F  2B         99          DEC HL
0040  73        100          LD (HL),E
0041  2B        101          DEC HL
0042  1A        102          LD A,(DE)
0043  77        103          LD (HL),A
                104  ;
                105  ;
                106  ;
                107  ;       SCANSIONE DISPLAY E TASTIERA
                108  ;       QUESTO SOTTOPROGRAMMA
                109  ;       CORRISPONDE AL MAIN DEL PROGRAMMA VISTAST
                110  ;
0044  21F00B    111  LOOP1   LD HL,BUFF
0047  7D        112  LOOP2   LD A,L
0048  FEF4      113          CP MAXBUF
004A  C8        114          RET Z
004B  0F        115          RRCA
004C  0F        116          RRCA
004D  0F        117          RRCA
004E  E660      118          AND COMASK
0050  4F        119          LD C,A
0051  7E        120          LD A,(HL)
0052  E60F      121          AND LOMASK
0054  CD6800  R 122          CALL PER
0057  7E        123          LD A,(HL)
0058  E6F0      124          AND HIMASK
005A  0F        125          RRCA
005B  0F        126          RRCA
005C  0F        127          RRCA
005D  0F        128          RRCA
005E  CBE7      129          SET 4,A
0060  CD6800  R 130          CALL PER
0063  2C        131          INC L
0064  18E1      132          JR LOOP2
                133  ;
                134  ;
                135  ;
                136  ;       SERVE NMI COME BREAKPOINT
                137  ;
0066  18B0      138  NMI     JR BREKEX
                139  ;
                140  ;
                141  ;
                142  ;       VISUALIZZAZIONE DI UN CARATTERE SU DISPLAY
                143  ;       VERIFICA SE E' PREMUTO UN TASTO
                144  ;       FORMA IL CODICE CORRISPONDENTE
                145  ;
0068  E5        146  PER     PUSH HL         ;SALVA PUNTATORE A BUFF
0069  21000C    147          LD HL,CONADD    ;INDIRIZZO CONSOLLE IN HL
006C  81        148          ADD A,C         ;FORMA SCAN + CAR
006D  77        149          LD (HL),A       ;USCITA
006E  E670      150          AND SCMASK      ;MASCHERA
0070  0F        151          RRCA            ;E SCALA SCAN
0071  0F        152          RRCA
0072  47        153          LD B,A
0073  CDE700  R 154          CALL DELAY
0076  CB7E      155  TEST1   BIT STB,(HL)    ;TEST TASTO PREMUTO
0078  2011      156          JR NZ ENDPER    ;NO, CONTINUA SCANSIONE
007A  7E        157          LD A,(HL)       ;SI, LEGGE IL PERIFERICO IN A
007B  CB7E      158  TEST2   BIT STB,(HL)    ;ATTENDE IL RILASCIO DEL TASTO
007D  28FC      159          JR Z TEST2
007F  E683      160          AND INMASK
0081  80        161          ADD A,B         ;FORMA IN A COD + SCAN
0082  CB67      162          BIT 4,A         ;VERIFICA SE DATO O COMANDO
0084  2007      163          JR NZ COMM      ;E' COMANDO: LO ESEGUE
0086  21F00B    164          LD HL,BUFF      ;E' DATO
0089  ED6F      165          RLD             ;AGGIORNA BUFF
008B  E1        166  ENDPER  POP HL
008C  C9        167          RET
                168  ;
                169  ;
                170  ;
                171  ;       CALCOLO INDIRIZZO DEL COMANDO
                172  ;
                173  ;
008D  E607      174  COMM    AND CDMASK      ;MASCHERA IL CODICE DEL COMANDO
008F  4F        175          LD C,A          ;MOLTIPLICA PER 3
0090  87        176          ADD A,A
0091  81        177          ADD A,C
0092  4F        178          LD C,A
0093  0600      179          LD B,0
0095  2AF80B    180          LD HL,(COMTAB)  ;INDIRIZZO INIZIALE TABELLA
0098  09        181          ADD HL,BC       ;INDIRIZZO DEL COMANDO IN HL
0099  E3        182          EX HL,(SP)      ;INDIRIZZO NELLO STACK E RIPRISTINA HL
009A  C9        183          RET             ;SALTA AL COMANDO (TABELLA COMSTA)
                184  ;
                185  ;
                186  ;
                187  ;       TABELLA SALTI ALLA
                188  ;       ESECUZIONE DEI COMANDI
                189  ;
009B  C3BE00  R 190  COMSTA  JP LAH
009E  C3B900  R 191          JP LAL
00A1  C3B700  R 192          JP EXM
00A4  C3B300  R 193          JP DEP
00A7  C3C900  R 194          JP GOTO
00AA  C3DF00  R 195          JP BREKSET
00AD  C3C300  R 196          JP E800
00B0  C3C700  R 197          JP DCR
                198  ;
                199  ;
                200  ;
00B3  3AF00B    201  DEP     LD A,(BUFF)
00B6  12        202          LD (DE),A
                203  ;
                204  ;       DOPO OGNI DEP ESEGUE UN EXM PER INCREMENTARE IL PUNTATORE
                205  ;
00B7  13        206  EXM     INC DE
00B8  C9        207          RET
                208  ;
                209  ;
00B9  3AF00B    210  LAL     LD A,(BUFF)
00BC  5F        211          LD E,A
00BD  C9        212          RET
                213  ;
                214  ;
00BE  3AF00B    215  LAH     LD A,(BUFF)
00C1  57        216          LD D,A
00C2  C9        217          RET
                218  ;
                219  ;
00C3  110008    220  E800    LD DE,800H
00C6  C9        221          RET
                222  ;
                223  ;
00C7  1B        224  DCR     DEC DE
00C8  C9        225          RET
                226  ;
                227  ;
00C9  31E20B    228  GOTO    LD SP,SAVSP+2   ;RIPRISTINA I REGISTRI DALLA TABELLA SAVREG
00CC  FDE1      229          POP IY
00CE  DDE1      230          POP IX
00D0  E1        231          POP HL
00D1  33        232          INC SP          ;SALTA SAVDE
00D2  33        233          INC SP
00D3  C1        234          POP BC
00D4  F1        235          POP AF
00D5  ED7BE00B  236          LD SP,(SAVSP)   ;RIPRISTINA STACK DI UTENTE
00D9  D5        237          PUSH DE         ;INDIRIZZO DI SALTO NELLO STACK
00DA  ED5BE60B  238          LD DE,(SAVDE)   ;RIPRISTINA DE
00DE  C9        239          RET             ;SALTO AL PROGRAMMA DI UTENTE
                240  ;
                241  ;
                242  ;
                243  ;       POSIZIONAMETO DEL BREAKPOINT
                244  ;
00DF  1A        245  BREKSET LD A,(DE)       ;SALVA IL CONTENUTO DELLA CELLA
00E0  32FA0B    246          LD (BRKSAV),A   ;DI BREAK
00E3  3EDF      247          LD A, BRKRST    ;SOSTITUISCE CON RST 2
00E5  12        248          LD (DE),A
00E6  C9        249          RET
                250  ;
                251  ;
                252  ;
                253  ;       RITARDO PER ANTIRIMBALZO
                254  ;
00E7  C5        255  DELAY   PUSH BC
00E8  0680      256          LD B, DELY      ;CARICA IL VALORE DEL RITARDO
00EA  10FE      257  HERE    DJNZ HERE       ;LOOP DI RITARDO
00EC  C1        258          POP BC
00ED  C9        259          RET
                260  ;
                261  ;
```

Figura 6 - Listato del programma PICOMON. 0.

Organizzazione del programma

PICOMON .0 (listato in Fig. 6) è costituito da un programma principale, che visualizza sul display il contenuto di una zona di memoria (BUFF) e gestisce la tastiera, e dai sottoprogrammi di esecuzione dei vari comandi. Un diagramma di flusso semplificato è riportato in Figura 7.
Durante l'inizializzazione viene predisposto (al valore di default BD0H) anche il buffer dello stack di utente (cella SAVSP). Questo perché l'esecuzione del comando di GOTO, che ripristina tutti i registri, richiede che lo stack di utente sia correttamente posizionato in RAM. Ciò é automaticamente garantito se si esce dal programma di utente con breakpoint (che salva tutti i registri, stack pointer compreso), ma può non essere vero per il primo GOTO dopo l'accensione del picocomputer.
Il funzionamento del loop principale é quello già descritto in VISTAST (BIT n° 3): premendo un tasto si genera nell'accumulatore il codice corrispondente alla combinazione riga/colonna del tasto premuto Figura 8a. Dato che nella nostra scheda i registri di I/O sono mappati su memoria, le istruzioni di IN e OUT di VISTAST sono sostituite da istruzioni di LoaD (LD). La coppia di registri HL é usata come puntatore al periferico (indirizzo CONADD).
Mappa della tastiera e codici dei tasti sono in Figura 8b. Una volta codificato il tasto, il programma determina se appartiene al campo dati (bit 4 = 0) o al campo comandi (bit 4 = 1). I nuovi dati sono introdotti in un buffer temporaneo, presentato sulle due cifre più a destra del visualizzatore; i comandi determinano un salto al corrispondente sottoprogramma di esecuzione. Questo salto ha luogo attraverso una tabella il cui indirizzo iniziale é definito in RAM (cella COMTAB), secondo la sequenza di operazioni di Figura 9. E' quindi possibile ridefinire le funzioni dei tasti nel campo comandi semplicemente alterando COMTAB e preparando un'altra tabella di salto, oltre ovviamente ai sottoprogrammi per l'esecuzione dei nuovi comandi.
Dopo l'esecuzione di un comando il controllo torna al monitor, tranne per il GOTO, che determina l'esecuzione del programma di utente.

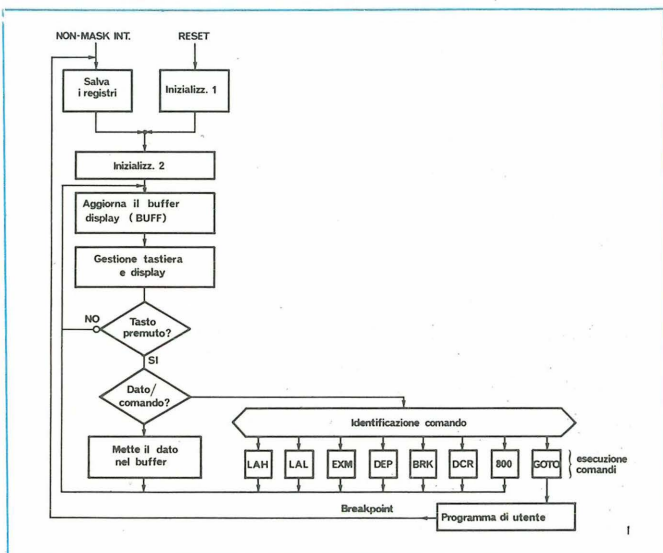



Figura 7 - Diagramma di flusso di PICOMON. 0.

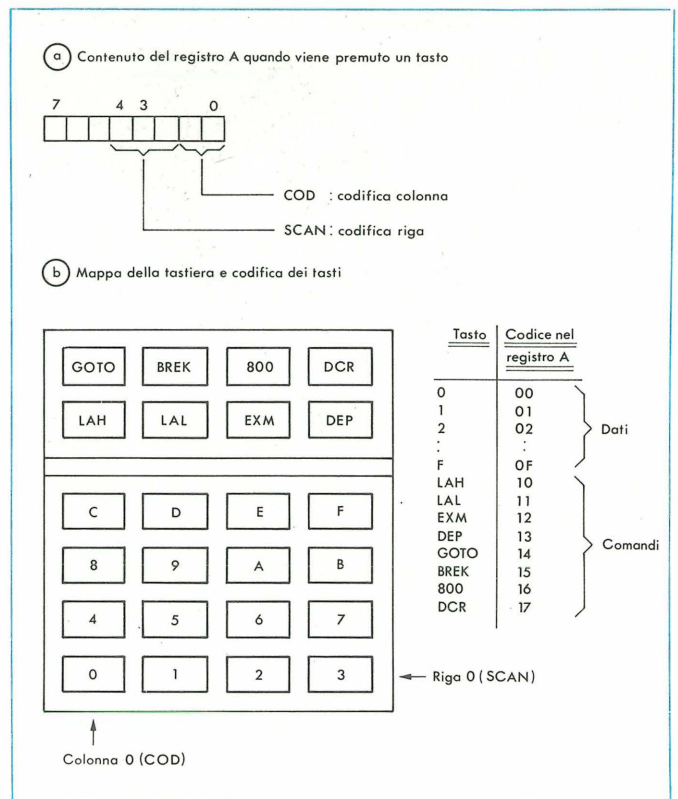



Figura 8 - Mappa della tastiera e codifica dei tasti.

Il visualizzatore di 8 cifre presenta Figura 10:
- un campo *indirizzo* (ADD), in cui compare, in codice esadecimale, il puntatore di indirizzo generato dal monitor (4 cifre).
- un campo *dati* (DATA), in cui compare il conte-

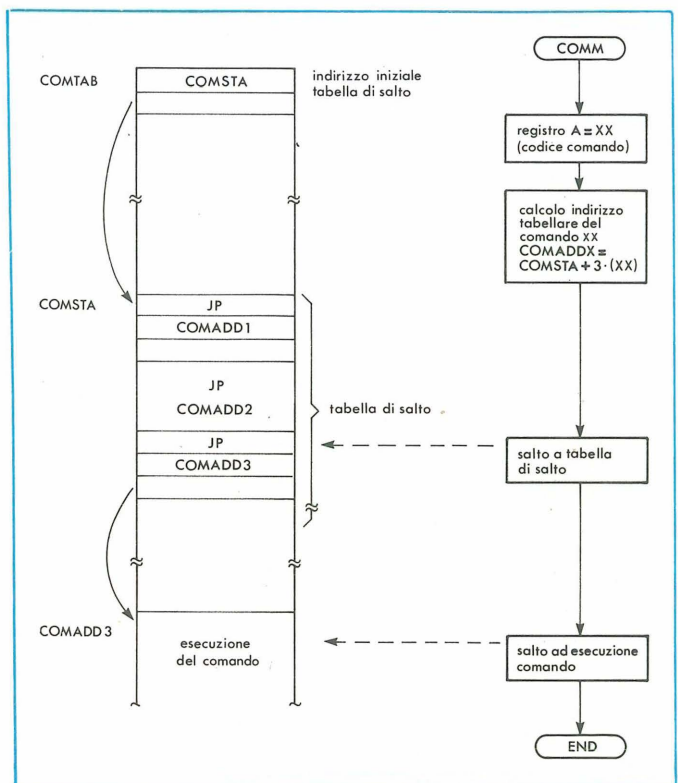
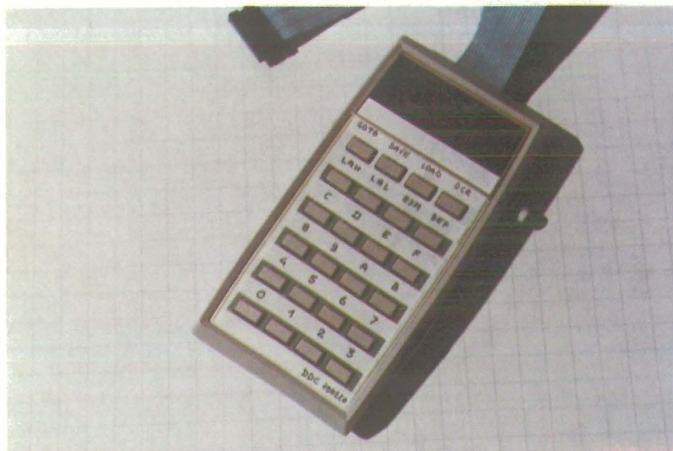



Figura 9 - Salto indiretto all'esecuzione dei comandi.

nuto della cella indirizzata dal puntatore ADD (2 cifre esadecimali)
- un campo *tastiera* (KEY), in cui compaiono, con scalamento automatico a sinistra, le ultime due cifre esadecimali introdotte dalla tastiera.

Durante l'esecuzione del monitor la situazione del display é sempre quella di Figura 10. Puntatore alla cella aperta é la coppia di registri DE. I comandi che modificano il puntatore intervengono direttamente su DE e solo successivamente viene aggiornato il buffer di display (sottoprogramma COP).

In Figura 11 è riportata la mappa della memoria RAM residente sulla scheda CPU (1 Kbyte da 800 a BFF).

Uso dei comandi di PICOMON. 0

Questo monitor consente di:
- leggere e scrivere su celle di memoria
- eseguire programmi di utente
- inserire breakpoints nei programmi di utente
- verificare, ed eventualmente modificare, dopo un breakpoint, i registri della CPU.

Le funzioni dei tasti del campo comandi della console sono:

LAH (*Load Address High*): carica il contenuto del campo KEY, cioé le ultime due cifre introdotte dal campo dati della tastiera, nel campo ADDH (cifre più significative del puntatore di indirizzo, vedi Figura 10).

LAL (*Load Address Low*): carica il contenuto del campo KEY nella parte bassa del puntatore di indirizzo (campo ADDL in Figura 10).

Questi due comandi predispongono il puntatore di indirizzo su una determinata cella. La cella così individuata é detta *aperta*, in quanto é possibile esaminare e modificare il suo contenuto.

Si può aprire una cella dando prima LAH e poi LAL o viceversa.

EXM (*EXaMine*): incrementa di una unità il puntatore ADD per aprire la cella successiva.

DEP (*DEPosit*): scrive nella cella aperta il contenuto del campo KEY ed incrementa il puntatore per esaminare la cella successiva.

DCR (*DeCRement*): decrementa il puntatore; é utile per verificare il risultato di una operazione di DEP appena eseguita.

Questo gruppo di comandi permette di scrivere nella memoria RAM i programmi di utente e di verificare il contenuto di celle di memoria qualsiasi.

Esiste ancora un comando diretto al puntatore:

800 Apre direttamente la locazione 800 (Inizio della RAM di utente); è utile e comodo perché quasi tutti i programmi scritti per il picocomputer partono dalla cella 800.

Gli altri comandi controllano l'esecuzione del programma di utente:

GOTO Determina l'esecuzione del programma che inizia alla cella aperta quando viene dato il comando (indirizzo che compare in ADD). I registri del banco principale vengono inizializzati dalla tabella SAVAREA.

BREK Inserisce un breakpoint (punto di interruzione dell'esecuzione e ritorno al monitor) all'indirizzo correntemente aperto. Quando, durante l'esecuzione del programma, si passa dalla cella ove é stato inserito il breakpoint, il controllo ritorna al monitor ed i registri del banco in uso vengono trasferiti nelle celle indicate nella tabella II.

Dopo l'esecuzione di un break compare sul display l'indirizzo di breakpoint ed é possibile esaminare il contenuto dei registri leggendo le celle di memoria specificate nella tabella con le normali procedure del monitor (apertura della cella; incremento o decremento del puntatore, ecc). Prima di riprendere l'esecuzione é possibile modificare il contenuto dei registri semplicemente modificando le corrispondenti

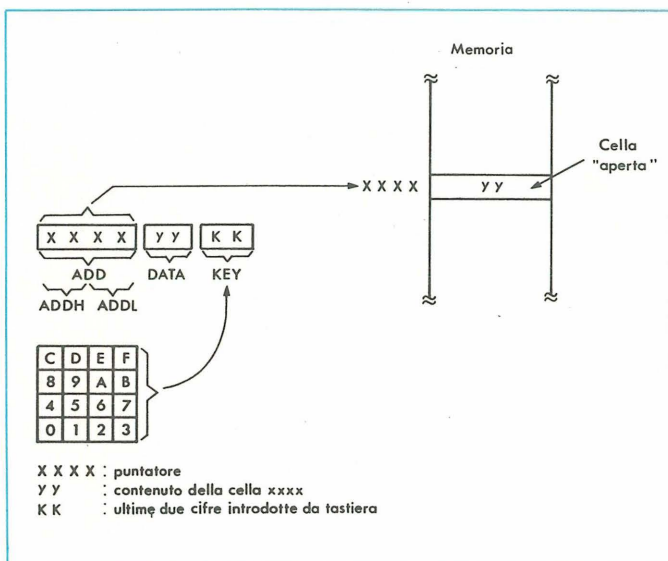



Figura 10 - Campi del visualizzatore e loro contenuto.

Abbiamo ricevuto numerosissime richieste sul Picocomputer e stiamo rispondendo a tutti.
Il Picocomputer è offerto nella configurazione H/S finora descritta, in Kit oppure assemblato e collaudato ad un prezzo che è intorno alle 150.000 lire. E' coperto da garanzia e da assistenza. E' fornito inoltre un manuale di montaggio e d'uso.
Per ulteriori informazioni scrivete, non telefonate.

celle di memoria. Solo i registri del banco in uso sono portati in RAM e possono essere modificati. Per continuare si usa il comando di ı GOTO, come prima descritto.
Il comando BREK opera sostituendo il contenuto della cella aperta con l'istruzione RESTART quindi possibile inserire il breakpoint solo su celle di RAM che contengano il primo byte di una istruzione. L'esecuzione del breakpoint ripristina il contenuto della cella. E' anche possibile inserire più di un breakpoint ma in questo caso non viene ripristinato correttamente il contenuto delle celle modificate.

Avvertenza

PICOMON. 0 (così come VISTAST), é estratto da un programma di circa 1 Kbyte, con altre funzioni, troppo lungo per essere presentato e spiegato in un solo articolo. Il programma complessivo é in uso già da tempo e quindi probabilmente corretto questo non garantisce però che parti di esso siano complete e funzionanti. Ad esempio nel listato di VISTAST (BIT n° 3, pag. 40), mancava la inizializzazione dello stack. Pur cercando di porre la massima attenzione nel preparare gli "estratti", non possiamo garantire la totale correttezza dei programmi presentati a titolo di esempio. Segnalazioni di eventuali errori sono gradite ed utili. ■

Tabella I - Uso della memoria RAM residente sulla scheda CPU (1 Kbyte da 800 a BFF)

Indirizzo	Nome	Funzione
BFF	—	Non usate
BF9/8	COMSTA	Indirizzo iniziale tabella comandi
BF7/6/5	INTSER	Libere per salto al programma di servizio dell'interruzione (da RST 7)
BF4	—	Non usata
BF3/2/1/0	BUFF	Buffer del display
BEF-BEO	SAVAREA	Area di salvataggio dei registri dopo un breakpoint
BEO	MOSP	Stack di monitor
BDO	SAVSP	Stack di utente (default)
800....		Disponibile per i programmi di utente

Tabella II- Mappa di SAVAREA

Indirizzo	Registri salvati
BEF/E	Program counter
BED/C	AF
BEB/A	BC
BE9/8	DE
BE7/6	HL
BE5/4	IX
BE3/2	IY
BE1/0	Stack pointer

Professionalità (all'italiana)

Poco meno di una ventina di anni orsono, quando l'Italia era invasa da stimoli innovativi sul piano tecnologico e la nostra industria elettronica si andava imponendo, seconda a poche sullo scenario europeo, facendo conoscere produzione e tecnologia di qualità sempre migliori, erano molti coloro che sostenevano che il "divario" tecnologico esistente tra l'Italia e gli Stati Uniti nel settore dell'elettronica sarebbe stato colmato entro breve tempo.

Questa previsione non si è verificata. Semmai il divario è aumentato e di "gap tecnologico" non se ne parla più da tempo. È stata una battaglia persa, e quindi si preferisce dimenticarla.

Ma perché?

Certamente sono molte le ragioni che hanno portato l'Italia a questo mancato successo e una loro approfondita analisi non è semplice. Sarebbe auspicabile che questi problemi venissero ripresi oggi e dibattuti in modo costruttivo su queste colonne. Poiché siamo stati chiamati a dire la nostra riteniamo che l'argomento, da noi liberamente scelto, possa essere di un certo interesse, vorremmo cominciare ad esaminarne una di queste molte ragioni di mancato successo che, anche se non di macroscopica evidenza, riteniamo possa aver pesato e continui ad influire in modo negativo sullo sviluppo tecnologico del nostro Paese.

In genere si tratta di "scarso livello di professionalità", a tutti i livelli.

Abbiamo ragazze che escono dalle scuole professionali, trovano un impiego e qui si scopre che non sanno tenere un'archivio in ordine perché non conoscono l'alfabeto e quando c'è da collocare lettere come K, J, Y o W non sanno dove mettere le mani.

Abbiamo periti di Istituti Tecnici che non sanno saldare e non hanno mai maneggiato un circuito integrato: non c'è da meravigliarsi se costoro potranno trovare difficilmente un posto di lavoro. Abbiamo dei periti in informatica che non hanno mai visto un terminale e quindi non hanno mai avuto l'opportunità di verificare un programma elaborato a tavolino, abbiamo studenti che non sanno studiare, insegnanti che non sanno insegnare. Quasi uno sfacelo.

Sono cose che tutti sappiamo e delle quali parliamo con aria rassegnata. In generale concludiamo stringendoci nelle spalle e dicendo: "la responsabilità non è mia: la colpa è del sistema". E il problema è accantonato ancora una volta: a tutti i livelli. Se facessimo un esame di coscienza dovremmo concludere che ben poco facciamo, a livello di ogni singolo individuo, per migliorare questo stato di cose. E, anche se sommessamente, dovremmo vergognarcene.

Abbiamo anche dei geni, è vero; ma purtroppo rimangono casi isolati. Ci illudiamo di essere una fucina di geni e ci riteniamo esportatori di buoni cervelli. In effetti non li esportiamo: sono loro che fuggono, perché restando qui non avrebbero le possibilità di dare una seria prova delle loro capacità. Se ne vanno altrove, là dove l'ambiente in cui si inseriranno li accoglierà nel modo più opportuno, riservando alle loro doti intellettuali ed organizzative quel livello di "rispetto professionale" che si meritano e che qui non viene loro riconosciuto, vuoi per gelosie di parte o più semplicemente per leggerezza o superficialità.

La mancanza di professionalità è certamente uno dei nostri maggiori problemi.

La superficialità collettiva con la quale siamo stati educati (o diseducati) ad affrontare i problemi pratici da una scuola non adeguata al momento attuale, è presente in ogni nostra azione. L'approssimazione, il compromesso, la superficialità professionale diventano così nostre caratteristiche: ci abituiamo così a fare i lavori in qualche modo e se il lavoro non è fatto bene non importa: ci penserà qualcun'altro a provvedere. In nome di un preteso rinnovamento culturale che tende a svilire ogni vero tentativo di educazione tecnologica avanzata a vantaggio di un'educazione

sempre più politicizzata e di parte, ci ritroviamo oggi la scuola nello stato in cui è, quasi assolutamente incapace di formare tecnici adeguatamente preparati. Naturalmente tendiamo a trovare altrove le responsabilità di questa scarsa preparazione culturale: per esempio diciamo che non ci sono soldi e che quindi non si può investire più di quel tanto nell'istruzione professionale. Tutti sanno quanto ciò non sia vero: si dica piuttosto che non si sanno spendere bene i soldi che ci sono. Forse un difetto che ci deriva dal tipo di educazione che ci viene impartita è quello di non attribuire al denaro il suo giusto valore e quindi siamo spesso portati agli investimenti sbagliati con conseguenti distruzioni di enormi capitali e considerevoli perdite di tempo.

Non sappiamo essere, in tutte le nostre decisioni quotidiane individuali, dei veri professionisti: questo è il punto. A parole sappiamo fare tutto, anche quello che gli altri non sanno fare, perché di parlatori ne abbiamo tanti e tutti ottimi; ma buoni "esecutori" ne abbiamo assai pochi. E invece è proprio di queste persone di cui abbiamo bisogno: persone che non si limitano a pensare ma che poi realizzano.

E di queste persone la scuola non ne fornisce abbastanza. Che cosa fare, dunque? Potrebbe ciascuno di noi fare qualcosa per migliorare questa situazione così drammatica? Noi siamo convinti di sì. Abbiamo già molti esempi positivi di scuole o Istituti, fra i quali molti privati o gestiti da importanti aziende, che preparano allievi volonterosi con un grado di professionalità che nulla ha da invidiare a scuole di tipo anglosassone. Putroppo questi esempi sono pochi.

Viene quindi spontaneo chiederci se non è il caso di promuovere ulteriormente l'addestramento di tipo privato, contrapponendone l'efficienza al generale disinteresse e povertà organizzativa di tante scuole pubbliche. E non sarà neppure il caso di esercitare eccessivi controlli qualitativi sulle attività di queste scuole: una naturale selezione, basata sul rapporto costo/efficienza, verrà lentamente operata nel tempo da parte dei "clienti" stessi di queste scuole. Cosicché le scuole che notoriamente garantiranno un buon rapporto costo/efficienza, vedranno salire le loro iscrizioni e quindi avranno buone probabilità di ulteriori investimenti per migliorare ancora questo rapporto, mentre le scuole che non potranno soddisfare il pubblico in questo senso, saranno destinate ad un declino più o meno rapido, ma inesorabile. Sarebbe anche auspicabile che queste scuole si mettessero in libera concorrenza con le scuole pubbliche in modo di dare il via ad una spirale di competizione qualitativa. Un pubblico culturalmente più evoluto sarebbe certamente in grado di operare scelte che potrebbero contribuire ulteriormente allo sviluppo di una cultura tecnologica della quale oggi non si può derogare se non correndo il rischio di pericolosi passi indietro sul nostro fronte socio-economico.

È chiaro che sono in gioco gli interessi di un'intera collettività, e quindi di ciascuno di noi. Sta quindi a noi individui darci da fare per collaborare su un progetto evolutivo di cultura tecnologica: ciascuno di noi si imponga di ragionare e di agire in modo "professionale", collaborando coi propri simili, e mettendo al bando l'approssimazione e l'arte di "arrangiarsi" alla quale sembriamo tanto affezionati. Lo sviluppo di una cultura tecnologica è anche un fatto di educazione: a questo la scuola può contribuire in una certa misura: il più lo deve fare ogni singolo individuo imponendosi di comportarsi secondo i canoni di un'etica altamente professionale, senza la quale sarà ben difficile per noi tutti reggere in futuro il confronto coi paesi tecnologicamente più progrediti.

Tutto ciò non sono che considerazioni e proposte. Tribuna è aperta a tutti.

Vogliamo iniziare un dibattito serio, critico e costruttivo, basato su fatti e non su parole per dare un contributo personale ed individuale all'evoluzione di una adeguata cultura tecnologica?

dr. Giuseppe Fontana

Corso sul PASCAL
Breve storia dei principali linguaggi

di **E. Waldner** - *Università di Bari, Istituto di Fisica*

Premessa

Il PASCAL si sta imponendo con prepotenza: forse è la più grossa novità venuta fuori nel campo dei linguaggi per calcolatori negli ultimi vent'anni. Si moltiplicano iniziative varie, dibattiti, richieste di corsi di aggiornamento. Sembra quasi che sia finito il lungo sonno del software, e che tiri aria nuova: il linguaggio universale, buono per il micro e per il macro calcolatore, flessibile, compatto, semplice, pressoché universale. Insomma il linguaggio del 2000.

E' vero? Francamente non lo so. Certo è che il linguaggio è estremamente potente ed interessante; tanto che ho deciso di inserirlo in un corso di laboratorio per studenti di Fisica, nella fondata speranza che il PASCAL sia d'uso comune fra qualche anno, quando cioè gli attuali studenti entreranno nel mondo produttivo. Possibilmente - lo spero per loro - in un laboratorio di ricerca, di cui tanta carenza c'è in Italia. Insomma è un scommessa. Però penso che ne valga la pena.
La presente serie di articoli vuole essere un'iniziativa nuova: vuole dare ai lettori un vero e proprio corso di auto-istruzione nella fondata convinzione che in un mondo tecnologico occorre imparare continuamente. Anche *dopo* il diploma o la laurea. Anzi, *soprattutto* dopo.
Difetti, mancanze, poca chiarezza potranno certo esserci: sarò grato a tutti coloro che vorranno segnalarmeli e/o discuterne.

L'ENIAC, *E*lectronic *N*umerical *I*ntegrator and *C*alculator, fu il primo calcolatore elettronico. Si era nel 1943, e la tecnologia usata era quella dei tubi a vuoto.

La storia

E cominciamo con un po' di storia ad uso e consumo dei neofiti. Le "vecchie volpi" che magari hanno programmato sul 1401-IBM o (Dio non voglia) sul 704-IBM possono liberamente saltare più avanti.

A partire da questo numero iniziamo la pubblicazione di una serie di articoli su Pascal. Ne abbiamo già parlato del n° 3 di BIT, e riprendere ora il discorso in maniera più sistematica risponde fra l'altro alla logica di entrare in merito a posizioni polemiche che si possono così riassumere:

1) *Pascal è un linguaggio molto "parlato" ma poco usato.*
2) *Non essendo stato commissionato da grosse industrie dell'informatica o da settori dello stato USA, non giungerà mai ad una standardizzazione come Fortran o Cobol.*
3) *È facile da usare per persone abituate al lavoro di tipo scientifico, ma è ostico per chi non ha una solida preparazione specifica.*
4) *Serve per grossi programmi ma non per piccole applicazioni, e inoltre genera programmi con bassa velocità di esecuzione, in quanto viene interpretato.*

Di queste critiche alcune denotano poca informazione, altre sono riconducibili alle posizioni di quelle industrie che, avendo grossi quantitativi di software già sviluppato in altri linguaggi, esprimono valutazioni non proprio disinteressate.

Entrando comunque in merito, occorre dire che:

1) *Pascal dal '77 è in continua espansione su piccoli sistemi.*
   Persino delle case che disponevano già di linguaggi ad alto livello, peraltro simili al Pascal (Zilog, INTEL) sono state forzate dalle richieste degli utenti ad implementare compilatori Pascal.
2) *È proprio il fatto di non essere stato commissionato che ha permesso di progettare un linguaggio che non dovesse sottostare a costrizioni di qualsivoglia specie. Entro i primi mesi dell'80 dovrebbero uscire gli standards. Va comunque notato che anche un linguaggio come ADA, commissionato dal dipartimento della difesa USA, è un linguaggio Pascal-like.*
3) *Non è vero che sia difficile, anche se è vero che forza a rispettare delle regole rigide a livello di costruzione dei programmi. Ma questa è una nota positiva, in quanto obbliga ad acquisire delle metodologie di lavoro che ampliano e la qualità del prodotto e la produttività.*
4) *Se un certo modo di programmare è essenziale per grossi sistemi, è comunque, certamente, positivo anche per piccole applicazioni. Pur non essendo l'interpretazione, nella maggioranza dei casi, un fattore limitativo, va fatto notare che esistono anche molti compilatori che generano il linguaggio nativo.*

Per tutti questi motivi riteniamo che Pascal debba essere diffuso a tutti i livelli: qualunque sarà il linguaggio degli anni '80, sarà comunque un linguaggio derivato da Pascal. Con questa serie di articoli BIT si propone di svolgere un lavoro di educazione alla programmazione, che ci auguriamo venga recepito da tutti; anche da chi è alle prime esperienze di programmazione.

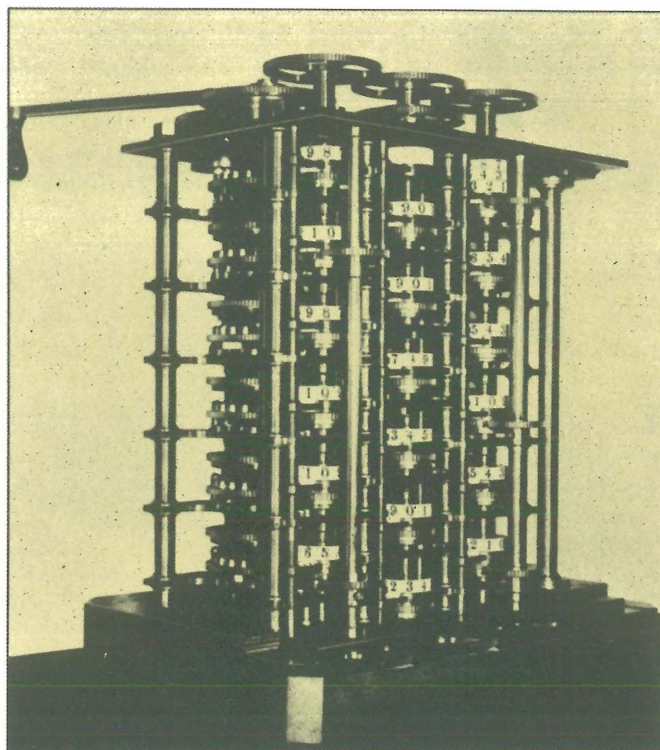
L'IBM 704 aveva una memoria centrale di 32 K words, ciascuna delle quali era di 36 bit. Pur essendo dotato di periferiche come lettore di schede, perforatore, stampante, unità magnetiche e display video, non realizzava la simultaneità di operazione di I/O e di calcolo interno.

Occorrerà andare con la memoria all'inizio degli anni 60, quando avvenne il gran balzo nei calcolatori: dalle valvole (IBM-704, 1956) ai transistors (IBM-7090, 1960). Era l'epoca del Cervello Elettronico, caro ai giornalisti. Si parlava molto nei circoli competenti (letterati, giornalisti, filosofi) se il calcolatore pensasse o no. Se la *Macchina* avrebbe soppiantato l'*Uomo.* Se ciò non preludesse ad un'era di generale disoccupazione. La realtà era ben diversa: i 12 μs di ciclo macchina della 704 ed i suoi 32 K di memoria facevano sì discettare sulla Macchina Pensante, ma si confronterebbero molto male oggi con un normale calcolatore da tavolo. Che però viene qualificato solo da Piccolo Mostro e non eccita la fantasia dei suddetti circoli competenti, preoccupati oggi di riversare la loro ben nota esperienza in campo energetico ed ecologico.

Erano anche i tempi in cui il calcolatore costava così tanto ed il software (peraltro molto elementare) così poco, che i produttori offrivano il software gratis, o quasi, a chi acquistava il calcolatore. Poi pian piano si è sviluppato il cosidetto "diagramma a forbice": sono crollati i costi dell'hardware, della ferramenta, e sono saliti i costi del software: programmazione, inventiva, specializzazione, complessità oggi si pagano carissime. E giustamente. A proposito della *Macchina* che soppianta l'*Uomo*.

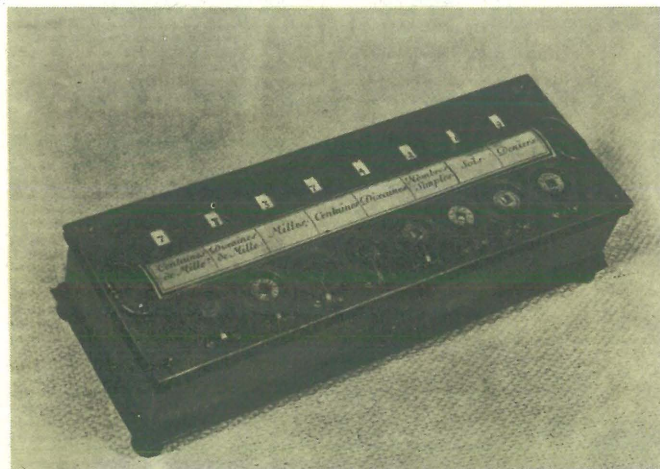
Fu in questo ambiente che si svilupparono i linguaggi simbolici. In effetti fino al 1958 (data di nascita del FORTRAN) la vita dell'utente era grama. Sempre prendendo a paradigma il 704-IBM (e la concorrenza era più o meno allo stesso livello) egli aveva a disposizione solo l'assembler, condensato in circa 100 paginette di manuale (comprese tabelle per la conversione ottale-decimale, etc.)

Praticamente contemporanei nacquero così due linguaggi: il FORTRAN e l'ALGOL, diversissimi per impostazione e per destini.

L'ALGOL infatti nacque essenzialmente a tavolino da teams di esperti e specialisti con il dichiarato scopo di essere *il* linguaggio per calcolatori. Era infatti un linguaggio completo, organico, con vaste possibilità. Aveva (ed ha) un solo difetto: creato per essere completo, risultò farraginoso. Chi, come me, ebbe la sfortuna di dover programmare con il calcolatore Olivetti ELEA-6001 in PALGO (*P*rogrammazione *AL-G*oritmica *O*livetti), diretta discendenza dell'ALGOL, ricorda ancora gli incredibili listings dei programmi, molto più simili ad opere letterarie che non a lavori tecnici. E ricorda - naturalmente -l'incredibile quantità di errori che in tali poemi si annidavano, renden-

La macchina delle sottrazioni di Babbage.

do la programmazione simile all'enigmistica. Se si aggiunge il fatto che le istruzioni erano su nastro perforato e che cambiarne una richiedeva di tagliare il nastro, eliminare il pezzo sbagliato, inserire quello nuovo saldandolo con lo Scotch (di Mylar, supersottile, altrimenti il fotolettore impazziva) si capirà facilmente di che tipo di lavoro si trattava.
La farraginosità giocò a scapito dell'ALGOL che venne praticamente snobbato da tutta la comunità scientifica.
Nessun fisico o chimico che io conosca ha mai osato aggredirlo. Diverso il discorso per gli informatici, e diverso anche il discorso per i calcoli "finanziari", dove il COBOL è - tutto sommato - in buona posizione.
L'avversione per le lungaggini e le complicazioni fu ed è così grande che neanche un cugino dell'ALGOL, il PL-1 ottiene oggi molta fortuna. Nato in casa IBM e ristretto a macchine IBM (e ciò è certo un pesante limite), il PL-1 si propose di essere più semplice e più potente del suo antenato. In effetti è vero. Però è usato da pochissimi, almeno fra coloro che usano il calcolatore per quello che è: un attrezzo per maneggiare tonnellate di dati e risolvere problemi altrimenti non aggredibili.
Diverso il destino per un altro cugino di questa famiglia: il PL-11. Nato in un ambiente di ricerca di particelle elementari (il CERN, laboratorio europeo a Ginevra) ha tutto l'impianto logico simile all'ALGOL, ma si rivolge ad un campo particolare: sostituisce l'assembler. E' insomma un cosiddetto "linguaggio a basso livello": non permette (salvo salti mortali) calcoli in virgola mobile, ma invece è possibile con esso operare su registri, stacks, interrupts etc con estrema disinvoltura. E' - per ora -un linguaggio limitato ai calcolatori Digital (da cui il nome: l'*undici* di PL-11 sta proprio per ricordare la serie dei PDP-11). Ma è un linguaggio che sta conquistando uno spazio notevole, nonostante i suoi attuali limiti. Probabilmente sarà l'embrione dei futuri assemblers: non più linguaggi del tipo "istruzione dopo istruzione", ma linguaggi simbolici che ad una poderosa struttura logica affiancano la possibilità di agire allo stesso livello degli assemblers senza le lungaggini di questi ultimi.
L'altro linguaggio ad alto livello ebbe una straordinaria fortuna. Nato con precise richieste di praticità, il FORTRAN fu fin dall'inizio semplice, essenziale e pratico. La versatilità e la completezza non furono però mai il suo forte. Nacque come attrezzo spiccio e senza troppe pretese: limitato nella logica (chi usa ancora i famigerati IF aritmetici?), pensate nell'input-output (I/O d'ora in avanti), faticosissimo da usare nel caso si volesse agire sulla parola o sul bit singolo, fu ed è il linguaggio di chi vuol fare un calcolo perchè ha bisogno dei risultati, e non gli importa poi molto se l'attrezzo che ha in mano è più o meno perfezionato. L'importante è che funzioni.
Questo atteggiamento commerciale e pragmatico, della pragmaticità ebbe però anche i vantaggi: il FORTRAN infatti è un linguaggio che si è evoluto e lo ha fatto direttamente sulle necessità degli utenti, soprattutto di quelli della comunità scientifica.
Partito semplice, essenziale, spiccio, si è via via arricchito di nuove possibilità fino a diventare notevolmente completo e ricco: a tal punto che oggi raramente ci si imbatte in una situazione che comunque non sia aggredibile in FORTRAN. E chi non sa usare il FORTRAN è costretto fatalmente a dipendere da altri anche per il più banale dei calcoli.
Pratico, ma pensato troppo poco, il FORTRAN soffre dei difetti opposti all'ALGOL. E' sintetico e veloce, ma ha dei difetti di struttura tali da ritenersi ormai giustificata l'idea che si stia avviando verso un'inesorabile agonia.
Un esempio (fra i tanti che si possono fare e che chi lavora con questo linguaggio conosce bene per averci sofferto) vi chiarirà ciò che voglio dire. Prendete il caso delle IF e supponete di dover eseguire il calcolo di Figura 1.
Il FORTRAN è già in crisi: dovremo infatti scrivere:

```
      IF (A.GT.0) GO TO 1
      B = 3
      C = 5
      GO TO 2
    1 B = 2
      C = 4
    2 CONTINUE
```

Né è possibile fare altrimenti. Infatti, se è vero che A > 0, abbiamo a disposizione *un solo* statement da poter eseguire, mentre invece a noi ne occorrerebbero due (B =2,C =4). Peggio ancora: dopo aver eseguito questo statement, il programma vorrebbe ripigliare

La calcolatrice di Pascal. Si osservi che il primo quadrante a destra ha 12 posizioni, e quello accanto 20, in quanto la macchina era destinata a fare calcoli sulla moneta francese in uso all'epoca.

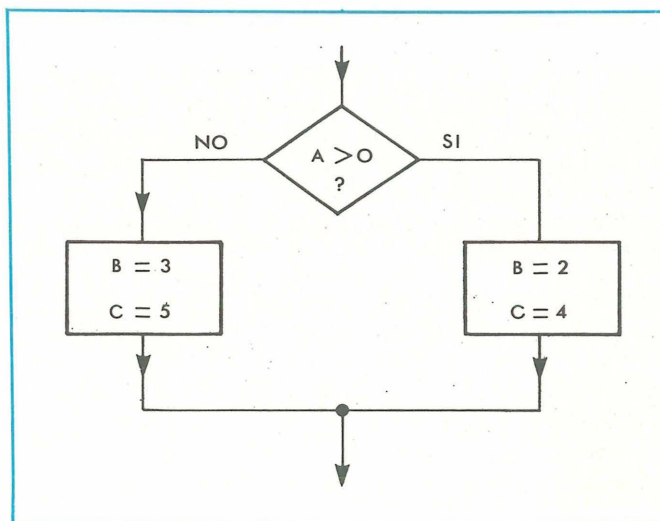



Figura 1

ad eseguire quello successivo, che sarebbe - guarda caso - ciò che dovrebbe fare se *non* fosse vera la condizione A > 0! Risultato? La logica a base di GO TO che abbiamo visto. E' l'unica scappatoia.
Con una logica un po' più complessa la cosa diviene ben presto infernale. Per capire qualcosa nei printouts ci si aiuta di solito con linee a matita che collegano i GO TO con i labels corrispondenti. Rapidamente un programma (soprattutto se non scritto da noi) diviene una selva di tratti paralleli, tale da meritare il nomignolo di "spaghetti-like programming".
E non è finita. Anche i calcoli iterativi in FORTRAN non sono così semplici come può sembrare: per esempio non è lecito effettuare iterazioni soggette a certe condizioni. In altri termini non si può dire "effettua queste istruzioni *fino a tanto che* è soddisfatta questa condizione". Anche in questo caso occorre operare in FORTRAN con una serie di IF e GO TO che esaltano la caratteristica programmazione a spaghetti.
Nonostante limitazioni di questo tipo (e ce ne sono molte altre) che peraltro affondano le loro radici nella struttura stessa del linguaggio e quindi non possono essere eliminate da un compilatore più intelligente o più perfezionato, il FORTRAN ha tuttora un uso diffusissimo. Potenziato da librerie estremamente vaste ed articolate con programmi di tutti i tipi, per statistica, calcolo integro-differenziale, minimizzazione di funzioni con un numero incredibile di variabili, routines grafiche, etc. etc., esso si presenta come l'assoluto dominatore nel campo della ricerca scientifica. Praticamente tutti i calcoli scientifico-tecnici vengono effettuati in FORTRAN da quasi vent'anni: anche se - occorre dirlo - la gente spesso brontola contro i salti mortali che è costretta a fare per riuscire a strizzare fuori dal linguaggio ciò che esso non potrebbe dare.
I limiti del FORTRAN non risultano mai tanto evidenti come quando occorre correggere un programma scritto da noi stessi o modificarlo, magari qualche mese dopo averlo steso. O - peggio ancora - quando occorre usare o modificare un programma scritto da altri. E' in questi casi che vien fuori tutta la pesante rozzezza della programmazione a spaghetti e tutti i limiti del linguaggio. Ci si difende di solito con statements di commento e con documentazione a parte. Ma basta che questa manchi o sia incompleta per una ragione o per l'altra e che la negligenza dell'autore gli impedisca di battere qualche statement di commento in più, perché un programma divenga quasi incomprensibile.
Morale: se è vero che il FORTRAN rappresenta bene l'antico adagio "val più la pratica della grammatica", è anche vero che alla fine chi vince è sempre la grammatica, magari dopo aver fatto tesoro dell'esperienza pratica.

Il PASCAL

Giungiamo così al PASCAL, il quale si propone di conciliare le due tendenze: di essere cioè un linguaggio completo e linguisticamente corretto e di essere semplice ed agile. Così come il PL-11, si propone di avere gli stessi usi di un assembler, ma con la potenza e la facilità d'uso di un linguaggio ad alto livello. Questi due saranno dunque i linguaggi del futuro? Non lo sappiamo. E' certo che non lo sarà il BASIC, in realtà un mini-FORTRAN la cui collocazione giusta è se mai nei calcolatori da tavolo.
Il PASCAL nacque nel '68, sulla linea dei linguaggi tipo ALGOL. Nel '70 fu scritto il primo compilatore e nel '73 si ebbe una seconda versione che per ora è rimasta più o meno immutata. Ha covato sotto la cenere per un po' di tempo e poi è stato scoperto sull'onda dei mini e dei micro (che adesso sono spesso micro solo di dimensioni). Oggi comincia a venir scoperto dalle grosse case. Texas, HP, Digital, CDC o lo hanno già implementato o si danno da fare, spinte dall'onda del consenso popolare. Non resta che attendere: le premesse per un successo ci sono, e sarebbe un successo che riscuoterebbe - oltre a tutto - vasti consensi.

Il BNF

Prima di parlare del PASCAL sarà necessario fare una premessa: il linguaggio è impostato in modo rigoroso e professionale e come tale va trattato. Ciò succede fin dagli inizi: poiché parlare di un linguaggio e descriverlo non è in genere facile, è stato deciso di adottare per la descrizione di PASCAL il formalismo di Backus-Naur (BNF d'ora in avanti) che è in uso già da tempo fra gli specialisti, ma non ha sempre raggiunto i livelli bassi della popolazione. Come noi.
Vediamo di che si tratta. Fu nel '59 circa che John W. Backus dell'IBM si pose il problema di come fare a parlare di un linguaggio senza dover per questo spendere mari di parole, spesso di difficile interpretazione. Fu nel '60 che prese forma un vero e proprio sistema di regole e formule (molto semplice, in realtà, come vedremo) con cui si può descrivere un linguaggio. Detto *B*ackus *N*ormal *F*ormalism in un primo tempo, venne poi detto *B*ackus *N*aur *F*ormalism per i contributi importanti che P. Naur dette al suo sviluppo e diffusione.
Il BNF è un linguaggio che *parla di* altri linguaggi. Ciò non è una novità nella logica matematica: questo tipo di attrezzo viene già da tempo chiamato *metalinguaggio* (vi ricordo che in greco *μετα* vuol dire "dopo, oltre, al di là") e per analogia le sue formule sono dette *metaformule,* i suoi simboli *metasimboli.* Un metalinguaggio usa quindi dei metasimboli organizzati in metaformule (o metaespressioni, a vostro piacere) per descrivere un linguaggio comune.
Le analogie con l'algebra sono evidenti: questa organizza dei simboli in formule per descrivere delle relazioni. Quando noi scriviamo:

$$S = \pi R^2 \qquad (1)$$

in realtà mettiamo in moto un complesso meccanismo che si basa sulle seguenti convenzioni:

I) con π indichiamo quel tale numero 3.14159 .... ottenuto in un certo modo
II) con R indichiamo un altro numero
III) con un 2 scritto in piccolo in alto a destra di R indichiamo l'operazione di elevamento al quadrato di R
IV) l'operazione di cui al punto III) dev'essere eseguita per prima
V) poiché fra π ed R non c'è nulla, si sottintende un'operazione di moltiplicazione

VI) il segno di = sta ad indicare la frase "si ottiene lo stesso numero che".

Insomma, detta a parole, la formuletta dell'area del cerchio suonerebbe così:
"se si misura l'area del cerchio si ottiene lo stesso numero che si otterrebbe se se ne elevasse al quadrato il raggio e si moltiplicasse il risultato per π, definito come il rapporto fra la lunghezza della circonferenza e quella del diametro di un qualunque cerchio".
E' piuttosto evidente che la (1) è molto più pratica e concisa
E ciò è vero anche del BNF. Vediamo anzitutto i vari metasimboli.

Metasimboli

< >: sono le parentesi angolari. Entro queste parentesi vanno messe le descrizioni (a parole) dell'elemento di linguaggio di cui si sta parlando. Scriveremo dunque

< programma > < frase > < costante >

ed intenderemo "un qualunque programma", "una qualunque frase", "una qualunque costante".
In un certo senso le parentesi angolari, con ciò che contengono, costituiscono l'elemento base del linguaggio: un po' come la lettera od il numero in algebra. Sono insomma delle *metavariabili.*
Come si lavora con le metavariabili? Le operazioni sono molto più semplici che non quelle dell'algebra. In effetti ce ne sono solo due, e sono la AND e la OR logiche. Le convenzioni sono che al posto della AND (che nel BNF acquista il preciso significato di "seguito da") non si scrive nulla, mentre al posto della OR si scrive la sbarretta verticale, |, al posto del simbolo + di solito usato nella logica booleana. Ciò corrisponde ad un'esigenza ovvia: si cerca fin dove è possibile di usare per i metasimboli qualcosa che non sia già sfruttato per i simboli del linguaggio che ci si accinge a descrivere.
Infine il simbolo relazionale che va letto "consiste di" o "è definito come", si scrive :: =, con ovvio riferimento al simbolo : = dell'ALGOL.
E qui praticamente finiscono le regole del BNF puro: si tratta ora di imparare ad usarle in pratica.

Metaformule

Vediamo alcuni esempi. Supponiamo di voler esprimere in BNF la seguente definizione FORTRAN:
"una subroutine consiste di un'intestazione, di statements dichiarativi e di istruzioni".
Scriveremo:

<subroutine>:: =< intestazione >
< statements dichiarativi >
< istruzioni > (3)

E qui ci troviamo di fronte al primo intoppo che ci obbliga ad allargare il metalinguaggio. In realtà noi sappiamo che la intestazione ha una forma particolare: è obbligatorio l'uso della parola SUBROUTINE che dev'essere seguita da un nome a piacere, il quale a sua volta *può* essere seguito da una coppia di parentesi, che contengono uno o più nomi di variabili o costanti (e se questi sono più d'uno, devono venire separati da virgole). Insomma ecco i vari casi:

SUBROUTINE TIZIO
SUBROUTINE CAIO (A) (4)
SUBROUTINE BEPPE (A,B,C)
SUBROUTINE GIGI (A (25,3),32,TONINO)

Come faremo ad esprimere tutto ciò in BNF? Posto che abbiamo già definito ciò che intendiamo come argomento di una subroutine (può essere una variabile, un elemento di matrice, una costante, una stringa alfanumerica ...), abbiamo comunque la necessità di ampliare il metalinguaggio nei seguenti modi:

I) ci occorre una convenzione qualunque per indicare delle parole o simboli in generale che sono esclusivi ed obbligatori nel linguaggio di cui stiamo parlando. Sono le cosiddette "reserved words", semplificate dalla parola SUBROUTINE nella (4).
II) ci occorre un qualche trucco per poter dire che una certa metavariabile può non esserci (cioè non è necessario metterla), può esserci ed essercene una sola, e può anche darsi che ce ne siano tante.

Cominciamo dal punto I): ci sono due tendenze: una di scrivere le parole obbligate (reserved words) fra apici (ed in tal caso uno scriverebbe 'SUBROUTINE') e l'altra che obbietta che l'apice è di solito un carattere frequente nei linguaggi e per evitare confusioni preferisce la sottolineatura: in tal caso si scrive SUBROUTINE intendendo che questa è una parola riservata, obbligatoria: insomma una parola del linguaggio. Io userò questa seconda alternativa che ritengo molto più pratica della prima e - soprattutto - molto più evidente per chi legge. E' evidente anche come mai le "reserved words" si chiamino anche *metacostanti.*
Va inoltre da sé che in BNF le metaformule possono occupare più di una riga e che è buona pratica cominciare ogni riga in modo "pulito" evitando di scrivere magari:

.... < espressione
aritmetica > (5)

Una scrittura pulita fa in fondo più parte del buon senso che non di regole del linguaggio ....
Per vedere come possiamo usare le parole riservate o metacostanti, possiamo prendere ad esempio la definizione:
"una costante intera è definita come un intero privo di segno oppure preceduto da uno dei due segni + o —"
Questa verrà scritta:

< costante intera >:: = < intero non segnato >
| + < intero non segnato >
| — < intero non segnato >

E così la defizione di ciò che si intende per digit numerico diviene:

< digit >:: = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 (7)

e quella di lettera diviene:

< lettera >:: = A | B | C | D | E | F | G | H
I | J | K | L | M | N | O | P
Q | R | S | T | U | V | W | X
Y | Z (8)

e quella di carattere alfanumerico:

< carattere alfanumerico >:: = <lettera> | <digit> (9)

Vediamo che la (7), la (8) e la (9) definiscono in modo completo il set di lettere e numeri e cosa si intende per carattere alfanumerico (che - come dice la (9) - è definito come una lettera da scegliere fra quelle della (8) o come un digit, da scegliere fra quelli della (9).

Più complicato il punto II) di cui sopra. Complicato ma essenziale, e quindi vale la pena di esaminarlo con cura.
Supponiamo che l'elemento A possa consistere o di niente (cioè non esserci), o di B, o di BB, o di BBB etc. Per esprimere una situazione di questo tipo si usa in genere una metaformula ricorrente scrivendo

< A > :: = < nulla > | < A > < B > (10)

Evidentemente ciò vuol dire che:

I) l'elemento A può non esserci (cioè "essere costituito da nulla")
II) Se l'elemento A c'è può essere costituito da B e da un altro elemento A. E qui delle due l'una: se questo secondo elemento A lo scegliamo come "costituito da nulla", vuol dire che non c'è, ed allora il primo elemento A è costituito solo da un B. Se però questo secondo elemento A lo scegliamo in modo che sia costituito da un solo B, allora il primo elemento A viene ad essere costituito da due B. Ed il gioco si può ripetere.

La (10) è bella ed elegante, ma ha un unico difetto: costringe a pensare. E siccome la gente, me compreso, è pigra, si è pensato bene di ricorrere ad un formalismo più semplice ed immediato. E' quindi invalso l'uso di un metasimbolo speciale per indicare che qualcosa può esserci, ma non necessariamente, e - se c'è - può essere ripetuto più di una volta.
Così si adottano parentesi quadre o graffe (a seconda dei testi) per dire la stessa cosa della (10). Potremo quindi scrivere

< A > :: = < nulla > | < A > < B >
< A > :: = [B]
< A > :: = {B} (11)

e tutte e tre indicano la stessa cosa. Noi non useremo mai la seconda alternativa, ma è bene sapere che in qualche testo la potrete trovare.
E' ormai in disuso la formula iterativa, usata solo dai puristi. Resta la terza alternativa, che è quella che useremo. Possiamo in questo modo definire ciò che intendiamo per numero intero:

< numero intero > :: = <digit> {<digit>} (12)

che si legge: "un numero intero consiste di un digit solo o seguito da altri digits" (e ciò che intendiamo per digit è definito dalla (7). Così la (12) definisce numeri quali:

3 25 14752 etc.

Anche una costante alfanumerica può venire definita in modo analogo:

< costante alfanumerica > :: = < lettera >
{ < digit >
< lettera > } (13)

che si leggerà: "una costante alfanumerica è definita da un insieme di numeri e/o lettere, il primo dei quali deve essere una lettera".
Il vantaggio delle parentesi graffe è anche un altro. E' invalso l'uso di attaccare degli indici alle parentesi; così:

$\{ B \}^{9}_{3}$ (14)

e leggere: "B è ripetuto *da un minimo di* 3 volte *ad un massimo di* 9 volte".
Con questa ulteriore estensione si può definire il nome di una variabile in FORTRAN, limitato - come sapete - ad un massimo di 6 caratteri alfanumerici, di cui il primo deve essere una lettera:

< nome variabile >:: = < lettera> { < lettera > |
< digit > $\}^{5}_{0}$ (15)

che vuol dire evidentemente: "il nome di una variabile è definito da una lettera seguita da zero fino ad un massimo di 5 lettere o digit numerici". Così ecco alcuni nomi di variabili leciti secondo la (15):

A
A23
B52C7 (16)
ZAX22K

ma *non* sono leciti

8XYZ (il primo carattere *non* è una lettera)
XYZABCD (ci sono più di 6 elementi) (17)

Il BNF è praticamente completo. Possiamo riassumerne quindi le regole e le definizioni in Tabella I.

< >	*parentesi angolari:* contengono la definizione di un elemento di linguaggio (metavariabile)
|	serve ad unire due metavariabili con un OR logica. La AND logica (da leggersi "seguito da") non si scrive, ma si scrivono semplicemente le due metavariabili una di seguito all'altra
——	*sottolineatura:* serve ad indicare le parole proprie del linguaggio (reserved words, o metacostanti)
:: =	*connessione:* sta ad indicare la frase "è definito come"
{ }	*parentesi di ripetizione:* indicano che gli elementi racchiusi all'interno possono essere ripetuti più volte: da un minimo di m volte ad un massimo di n volte.

Tabella I - Metasimboli del BNF

Alcuni esempi

Possiamo ora moltiplicare gli esempi: anche l'algebra può essere intesa come un linguaggio e possiamo descriverla col BNF.
Definiamo anzitutto degli operatori: l'operatore di moltiplicazione o divisione, che indicheremo con < m.d. >

< m.d. > :: = <moltiplicazione> | <divisione> (18)

e quello di somma algebrica <s.a.> che può essere una addizione od una sottrazione.

< s.a. > :: = < addizione > | < sottrazione > (19)

Un monomio può essere quindi definito come un numero reale legato ad un'eventuale parte letterale solo da operazioni di moltiplicazione e divisione: in BNF diviene:

< monomio > :: = < numero reale > { < m.d. >
< lettera >} (20)

ed un polinomio, che è una somma algebrica di monomi,

< polinomio > :: = < monomio > { < s.a. >
< monomio > } (21)

Le (18, 19, 20, 21) sono tutto ciò che ci serve per capire cosa intendiamo per monomio o polinomio. Messe così sono in realtà più che altro un'esercitazione accademica, però a mano a mano che ci addentreremo

nel linguaggio vedrete che il BNF permette di ridurre considerevolmente la sintassi, di evitare incomprensioni di vario genere oltre a dare fin dall'inizio un taglio professionale allo studio del linguaggio.
Vediamo ora un altro esempio, e parliamo di analisi logica: cerchiamo di dare il BNF di una proposizione. Qui ci troviamo di fronte ad una questione interessante in quanto nella definizione possiamo seguire *due* vie: definire prima di tutto ciò che intendiamo come proposizione, usando metavariabili che non sono ancora definite, ma che definiremo in seguito, *oppure* seguire la via inversa che consiste nel partire dalle metavariabili fondamentali e combinarle via via in definizioni sempre più complesse fino a raggiungere la definizione voluta. I due metodi prendono il nome di TOP-DOWN e DOWN-TOP (o BOTTOM-UP) rispettivamente, e - incidentalmente - corrispondono anche a due metodi di programmazione. Nel primo caso si parte dal problema "all'ingrosso" specificandolo via via, costruendo subroutines sempre più elementari, finché si arriva agli ultimi pezzettini. Nel secondo caso si formano prima i mattoni da costruzione e via via si fa crescere l'edificio. Chi ha ragione? Al solito nessuno: il secondo metodo è più adatto a menti analitiche ed il primo a menti sintetiche. Dopo lunghe sofferenze ho deciso di regolarmi come meglio mi sento: e sono felicissimo di questa decisione.
Una proposizione quindi può venir definita TOP-DOWN così:

Questa verrà scritta:

< proposizione > :: = < soggetto >
< predicato verbale >
{ < complemento> }

< soggetto > :: = < sostantivo > } < aggettivo > |
< sostantivo > }
< predicato verbale > :: =
< complemento > :: = < preposizione >
< sostantivo >
{ < aggettivo > }

e lascio a voi il divertimento di controllare che le metaespressioni qui sopra riproducono effettivamente ciò che sapete di analisi logica.
Incidentalmente, la definizione di polinomio era un esempio BOTTOM-UP.
Non sempre vale la pena di usare il BNF: ad esempio in linguaggi con sintassi molto semplice esso è sprecato. Niente di nuovo. Gli attrezzi potenti si usano quando servono: è inutile usare un TIR quando basta il carrello della spesa. Per esempio è inutile che vi esercitiate ad usare il BNF magari sull'assembler: vi trovereste di fronte a cose molto banali del tipo
< istruzione di MOVE > :: = MOV < blank > < simbolo >, < simbolo > e simili.
Questo però mi fa venire in mente che mi sono dimenticato di dirvi un paio di cose: la prima è che non sempre si sottolinea una metacostante: quando è costituita da *un solo* carattere non lo si fa.
La decisione ha semplici ragioni di chiarezza tipografica: provate a pensare di sottolineare una virgola (come era il caso per la MOV) e sarete d'accordo anche voi. La seconda è che qualora un blank sia necessario lo si scrive esplicitamente come metavariabile, preferendo scrivere MOV < blank > che non magari MOV, oppure MOV-. Anche qui le ragioni tipografiche (con annessa possibilità di errori di stampa) sono evidenti.
Vi consiglio di divertirvi a scrivere qualche metaformula sui linguaggi che conoscete. La pratica vi sarà molto utile quando cominceremo a trattare il PASCAL con il BNF, e tanto più sarete allenati, con tanto minore sforzo seguirete quanto verrà detto. Buon lavoro.

■

A Londra il quinto convegno Euromicro

L'annuale convegno sulla microprogrammazione e sui microprocessori ha avuto luogo a Göteborg (Svezia) il 28, 29 e 30 agosto scorso con la partecipazione di più di 450 specialisti del settore.
Il successo dei precedenti incontri di Nizza, Venezia, Amsterdam e Monaco è stato anche stavolta confermato da questo convegno che è il maggiore avvenimento scientifico dell'anno nel campo dei microprocessori e della microprogrammazione.
Hanno parlato cinquantadue relatori provenienti da tutti i paesi Europei oltre che dal Giappone e dall'America; 20 specifici temi sono stati approfonditi dai vari gruppi di lavoro.
Il prossimo convegno si terrà a Londra il 16, 17 e 18 settembre 1980 e la scheda di partecipazione è già disponibile. La sede dell'Euromicro si trova a Parigi (14/18 rue Planchet tel. 367.41.27 - telex 211801).

Ormai sono in molti a dirlo:

MMD1

L'unico microcomputer didattico che lavora con entrambi i microprocessori

MMD1-A, assemblato
£ 445.000+IVA
MMD1-K, in kit istruzioni in italiano £ 315.000+IVA

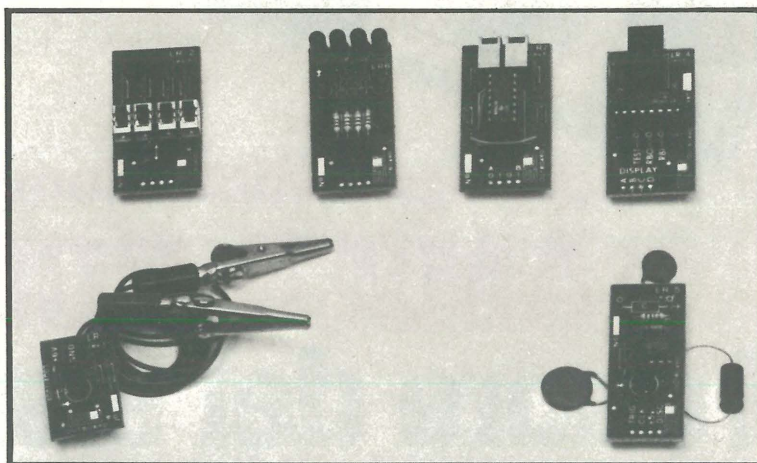
8080A e Z-80* ...

* con l'adattatore MMD1-Z80

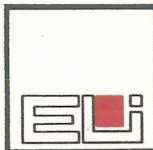
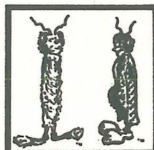
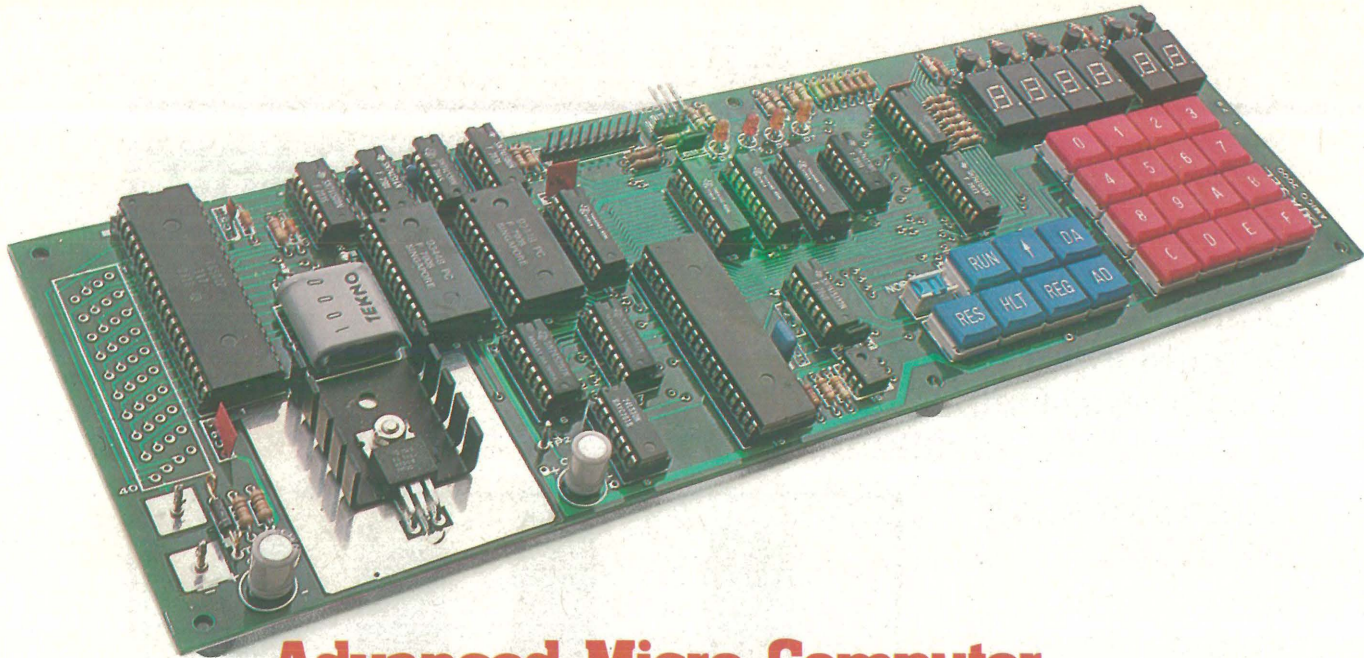
...e che dispone di OUTBOARD®

LR 4 - Display a 7 segmenti con decoder/driver
LR 27 - Octal Latch
LR 29 - General Input Port
LR 50 - Single Step Outboard
LR 25 - Outboard universale: comprende LR2, LR5, 2 LR6, LR7

Per la realizzazione dei
100 ESPERIMENTI
descritti e condotti
passo-passo nei famosi

BUGBOOKS
V° e VI°

Punti di vendita microcomputer MMD1, BUGBOOKS e sussidi didattici

10064	PINEROLO (TO)	(0121)	22444	**CAZZADORI** Via del Pino 38
10146	TORINO	(011)	773147	**GOMA ELETTRONICA** Via Valgioie 1
12051	ALBA (CN)	(0173)	49846	**CEM di A. Cania** Via S. Teobaldo 4
12100	CUNEO	(0171)	2773	**ELECTRONICS** Via Statuto 10a
13051	BIELLA (VC)	(015)	21070	**G. LANZA** V.le Matteotti 2/4
15076	OVADA (AL)	(0143)	821055	**ELTIR di S. Tirandi** P.zza Martiri della Libertà 30
15100	ALESSANDRIA	(0131)	443200	**GIOCO SCUOLA** Via Mazzini 36
16179	GENOVA	(010)	581254	**ELETTRONICA LIGURE srl** Via Odero 30
20131	MILANO	(02)	2894967	**FRANCHI CESARE** Via Padova 72
20156	MILANO	(02)	3086931	**AZ ELETTRONICA** Via Varesine 205
21013	GALLARATE (VA)	(0331)	797016	**ELETTROMECCANICA RICCI** Via Poscastello 16
21040	CISLAGO (VA)	(02)	9630511	**ELETTROMECCANICA RICCI** Via C.Battisti 792
21100	VARESE	(0332)	281450	**ELETTROMECCANICA RICCI** Via Parenzo 2
22100	COMO	(031)	507555	**SIRO di S. Rosean & C. sas** Via P.Paoli 47a
25100	BRESCIA	(030)	362304	**DETAS SpA** Via C. Quaranta 16
27036	MORTARA (PV)	(0384)	99960	**ZETA DUE AUTOMAZIONE** Via Beldiporto 14
28040	ARONA (NO)	(0322)	3788	**CEM di G. & C. Masella** Via Milano 32
31015	CONEGLIANO (TV)	(0438)	34692	**ELCO ELETTRONICA** Via Manin 26b
34133	TRIESTE	(040)	30341	**RADIO KALIKA** Via Cicerone 2
34170	GORIZIA	(0481)	32193	**B.E.S. di Bozzini & Sefcek** V.le XX Settembre 37
35100	PADOVA	(049)	654500	**ING. G. BALLARIN** Via Jappelli 9
36016	THIENE (VI)	(0445)	361904	**ELETTROACUSTICA VENETA** Via Firenze 24
38068	ROVERETO (TN)	(0464)	33266	**AGEC** Via Pasubio 68
40129	BOLOGNA	(051)	368913	**ZANIBONI ADRIANO** Via T. Tasso 13/4
41049	SASSUOLO (MO)	(059)	804104	**HELLIS di B. Prati** P.zza Amenodola 1
41100	MODENA	(059)	300303	**LART ELETTRONICA** Via Bellinzona 37a
71100	FOGGIA	(0881)	72553	**ATET di D. Fenga** Via L. Zuppetta 28
80125	NAPOLI	(081)	630006	**A.E.P. srl** Via Terracina 311
95128	CATANIA	(095)	447377	**RENZI ANTONIO** Via Papale 51

MICROLEM
20131 MILANO
Via Monteverdi 5
divisione didattica

Uffici commerciali
20131 MILANO, Via Piccinni 27
(02) 220317 - 220326 - 200449 - 272153
36016 THIENE (VI), Via Valbella cond. Alfa
(0445) 364961 - 363890
10122 TORINO, C.so Palestro 3
(011) 541686 - 546859

Advanced Micro Computer
AMICO 2000
Il cuore del sistema.

Un sistema completo a microelaboratore da autocostruire e tutto il supporto didattico necessario.

Caratteristiche

CPU: microprocessore 6502
Memoria RAM: 1kbyte
Memoria ROM contenente il Monitor
Tastiera esadecimale + tasti funzionali e passo singolo
Visualizzazione LED a 6 cifre
linee di ingresso e uscita parallelo
Generatore di clock quarzato
Regolatore di tensione incorporato
Alimentazione 5Vcc (non regolati), 800mA max.
Predisposto per l'espansione della RAM (1K)
Predisposto per l'interfaccia con registratore a cassette

Prezzi AMICO 2000A (IVA 14% esclusa)

- In scatola di montaggio Lit. 195.000
- Kit ER1 di espansione 1kByte RAM Lit. 25.000
- Kit EC2 per interfaccia registratore a cassette Lit. 30.000
- Versione montata e collaudata completa di espansione RAM e interfaccia cassette Lit. 285.000

Il sistema AMICO 2000 viene descritto in una serie di articoli didattici pubblicati sulla rivista SPERIMENTARE a partire dal numero di Dicembre 1978.

L'AMICO 2000 è progettato in Italia dalla

Via Cortina D'Ampezzo, 17
Milano - Tel. 02/5391719

Bit

Prego inviarmi senza alcun impegno da parte mia:

☐ Ulteriori informazioni sul sistema AMICO 2000
☐ Le modalità per l'ordinazione e il pagamento

Nome ____________ Cognome ____________

Via ______________________ N. ______

Città ____________________ C.A.P. ______

Implementazione software del single step e del breakpoint in un sistema con Z80

di **F. Luraschi** *SGS-ATES, Laboratorio Applicazioni Microprocessori, Milano*

Le prestazioni di breakpoint e di single step sono due caratteristiche fondamentali dei programmi di debugger. Le tecniche di implementazione sono sostanzialmente due: hardware e software. I vantaggi dell'approccio software.

Introduzione

Il debugger è un programma che fornisce all'utilizzatore un valido ed efficace strumento di collaudo e di verifica dei propri programmi.
Le più importanti operazioni che esso permette di svolgere sono due:
1) Visualizzazione e modifica dello stato della CPU, della memoria e delle periferiche.
2) Controllo dell'esecuzione del programma utente, consentendo di far partire il programma da un indirizzo prefissato (comando di GO), di interrompere l'esecuzione all'occorrenza di un dato evento (breakpoint) e ad ogni esecuzione di una istruzione (single step).

Scopo di questo articolo è descrivere la soluzione adottata per realizzare quest'ultima operazione nei debuggers MO-Z e NC-Z dei microcomputers NBZ80 e CLZ80 della SGS-ATES.
Questa soluzione, basata su alcune caratteristiche peculiari della CPU Z80, ha il vantaggio di non richiedere alcuna circuiteria oltre a quella necessaria a costituire il microcomputer e di ridurre di conseguenza il costo del sistema rispetto ad altre soluzioni.

Il breakpoint

Il breakpoint è in effetti la prestazione chiave di ogni debugger: se l'implementazione del breakpoint è corretta, la funzione di *single step* può infatti essere pensata come l'esecuzione automatica di una serie di breakpoints il cui indirizzo viene di volta in volta incrementato.
L'implementazione del breakpoint può essere fatta essenzialmente in due modi: mediante un hardware di supporto o con tecniche completamente software.
L'approccio hardware consiste praticamente in un comparatore ed in un registro di confronto, che viene caricato volta per volta dal debugger. Ha il vantaggio di una notevole flessibilità, in quanto qualsiasi combinazione di stati del bus indirizzi, del bus dati e dei segnali di controllo relativi può essere scelta dal programmatore come situazione di breakpoint.
Lo svantaggio consiste naturalmente nel costo dell'hardware addizionale, che nel caso di un single board computer raddoppierebbe praticamente il costo del sistema, e nel fatto che, se si vogliono più breakpoints attivi contemporaneamente, occorrerebbero tanti comparatori quanti sono i breakpoints che si vogliono realizzare.
Per contro l'approccio software, se non porta ad alcun costo addizionale, presenta lo svantaggio di funzionare solo su programmi residenti in RAM e di generare breakpoints *volatili*.

Per spiegare meglio che cosa si intende per breakpoint *volatile*, immaginiamo un programma contenente un loop a cento iterazioni di cui si vuole studiare il comportamento inserendo un breakpoint; nel nostro caso, il breakpoint verrebbe eseguito la prima volta, ma per le novantanove volte successive il loop verrebbe eseguito senza alcun controllo.
Si potrebbe rimediare all'inconveniente mettendo due breakpoints all'interno del loop ed approfittando di uno dei breakpoints per ristabilire l'altro, ma si tratta chiaramente di una procedura utilizzabile solo in casi di emergenza.
La ragione di questo malfunzionamento sta nella tecnica che i debuggers software usano per implementare i breakpoints, che consiste nel sostituire all'istruzione del programma utente dove si intende porre il breakpoint un salto, o, nel caso dello Z80, una restart al debugger.
Quando, eseguito il breakpoint, si vuole continuare nell'esecuzione del programma, il debugger è obbligato a rimettere al posto della restart l'istruzione originale (che è poi la prima da eseguire), e così facendo distrugge il breakpoint.

Nei debuggers MO-Z e NC-Z i breakpoints sono realizzati sostituendo all'istruzione specifica l'istruzione RST 38H.
Quando il programma utente raggiunge un breakpoint, salta alla locazione 38H, memorizzando nello stack pointer l'indirizzo del breakpoint.
Nella locazione 38H è memorizzato un salto alla routine SAVE del debugger, la quale provvede a memorizzare in una zona di memoria riservata al debugger e denominata *tavola dei registri* il contenuto di tutti i registri della CPU al momento del breakpoint.
Questo è possibile poiché le istruzioni di RST 38H e

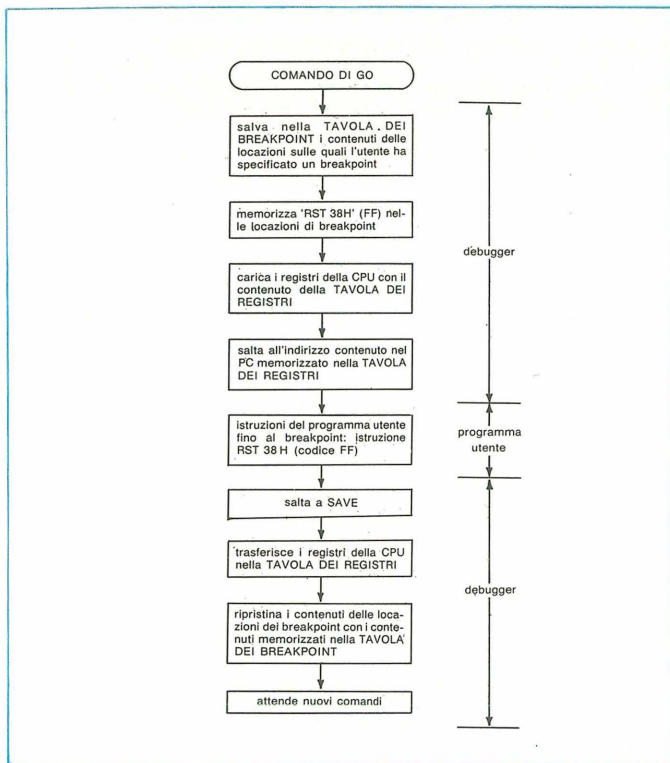



Figura 1 - Sequenza delle operazioni di debugger per il breakpoint.

JP SAVE non modificano alcun registro tranne il PC, il valore del quale, al momento del breakpoint, è però memorizzato nello stack dalla istruzione RST 38H.
Il cambiamento del codice della istruzione sulla quale è stato specificato un breakpoint con il codice di RST 38H non è visibile all'utente, poiché il debugger esegue questo cambiamento solo quando l'utente invia il comando di esecuzione del programma, memorizzando il codice dell'istruzione in una zona di memoria riservata, denominata *tavola dei breakpoints.*

Al verificarsi del breakpoint, il debugger, dopo aver eseguito il salvataggio dei registri della CPU, ripristina le locazioni sulle quali era stato specificato un breakpoint con il loro contenuto originale, memorizzato nella tavola dei breakpoints.
La Figura 1 illustra le operazioni compiute dal debugger da quando l'utente fa partire il programma con il comando di GO fino a quando viene raggiunto un breakpoint.

Il single step

Un altro potente strumento di debugging è fornito dalla possibilità di eseguire programmi in *single step.*

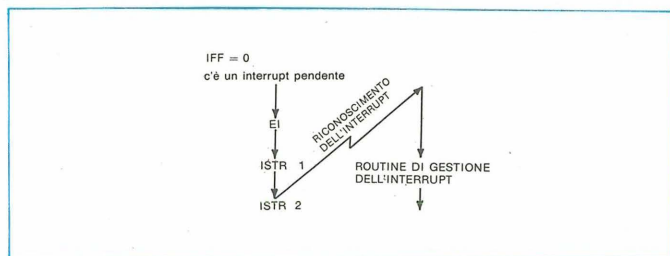



Figura 2 - Single step con interrupt vettorizzato.

Consiste nell'eseguire un numero specificato di istruzioni del programma utente, a partire da quella indicata nel PC memorizzato nella tavola dei registri (e più brevemente indicato con *PC utente*) e nel visualizzare i registri della CPU dopo l'esecuzione di ogni istruzione.
L'esecuzione in *single step* è, normalmente, utilizzata quando, nella fase di debugging, si è rilevata la presenza di un errore in una piccola zona di programma, senza riuscire, tuttavia, ad identificare, con precisione, la sorgente di tale errore.
In tale situazione è molto utile eseguire in single step le istruzioni appartenenti a questa zona di programma, in modo da vedere, in tutti i particolari, le azioni compiute da ciascuna istruzione, ed arrivare a scoprire la causa dell'errore più velocemente che utilizzando i breakpoints.

L'esecuzione di un programma in *single step* comporta, per ciascuna istruzione da eseguire, le seguenti operazioni:
1) Ripristino dei registri della CPU con il contenuto della tavola dei registri e salto alla locazione indirizzata dal PC utente.
2) Rientro in debugger (routine SAVE) dopo aver eseguito una istruzione del programma utente.
3) Salvataggio dei registri della CPU nella tavola dei registri
4) Visualizzazione della tavola dei registri.

Queste operazioni sono molto simili a quelle richieste per i breakpoints; le uniche differenze sono:
- La mancanza di memorizzazione del codice di RST 38H nelle locazioni di breakpoint specificate
- Il rientro in debugger dopo l'esecuzione di una istruzione invece che al verificarsi di un breakpoint.

Delle 4 operazioni necessarie a realizzare il *single step*, quella del rientro in debugger è l'unica che non può essere realizzata solo con il software.
Infatti è impensabile che il programma di debugger interpreti l'istruzione da eseguire, in modo da inserire un breakpoint sulla istruzione ad essa successiva, in quanto questa operazione richiede non meno di 1K byte di istruzioni per la Z80 CPU.
Un'altra soluzione è di utilizzare un circuito particolare che genera un NMI o un reset parziale della *Z80 CPU* dopo il *fetch* della istruzione del programma utente, con il conseguente svantaggio di un aumento del costo del sistema.
Un'altra soluzione che ha, però, il vantaggio di non richiedere circuiti particolari e che necessita di poche decine di istruzioni nel programma di debugger è quella di utilizzare un interrupt vettorizzato per il rientro in debugger.
Questa soluzione, adottata nei debuggers MO-Z e NC-Z, prevede per il rientro un interrupt *maskable* generato da un PIO, già presente nel sistema per altri scopi.
Presenta le seguenti limitazioni (che nella maggior parte dei casi non sono incontrate dall'utente nella fase di debugger):
- Il programma utente deve risiedere in RAM
- Le istruzioni IM0 e IM1 non devono essere eseguite in *single step.*
- L'istruzione da eseguire non può leggere il contenuto della locazione precedente.

La generazione dell'interrupt di rientro al momento

giusto è possibile utilizzando l'istruzione EI, con la CPU Z80 predisposta nel modo 2 di gestione degli interrupts (modo di interrupt vettorizzato).
Questa istruzione ha il duplice effetto di abilitare l'acquisizione degli interrupts da parte della CPU (IFF ←1) e di proteggere l'istruzione successiva dagli interrupts.
Questo significa che, se una richiesta di interrupt è generata prima della esecuzione della istruzione EI e la CPU ha l'acquisizione degli interrupts disabilitata (IFF = 0), con l'esecuzione della istruzione EI l'interrupt viene riconosciuto non al successivo *fetch* ma dopo aver eseguito l'istruzione seguente ad EI.
Questo fatto è illustrato in Figura 2.
Prima di saltare alla routine di gestione dell'interrupt, vengono eseguite completamente le istruzioni EI e ISTR 1.
Ritornando al *single step*, se immediatamente prima dell'istruzione da eseguire viene messa una istruzione di EI e viene passato il controllo a quest'ultima, l'interrupt di rientro viene generato dopo l'esecuzione dell'istruzione del programma utente.
In questo modo viene, però, eseguita oltre all'istruzione specificata anche l'istruzione di EI, e, pertanto, lo stato di tutti i registri della CPU, al rientro in debugger, risentirà dell'azione di entrambe le istruzioni e del fatto che il registro I, che costituisce la parte più significativa del vettore di interrupt, è posizionato dal debugger, prima dell'esecuzione ad un ben definito indirizzo (routine SAVE).

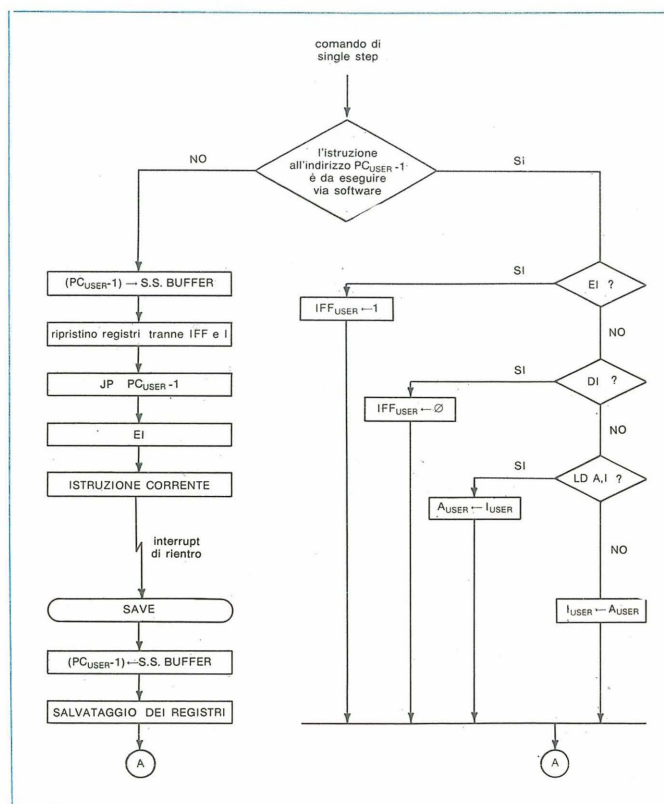
D'altro canto, questo fatto non invalida il metodo di realizzare il *single step* con un interrupt vettorizzato, poiché l'istruzione di EI esegue solo la semplice operazione di abilitare l'acquisizione degli interrupts da parte della CPU (IFF ← 1) e, inoltre, le istruzioni che operano con il registro I e il flip-flop IFF sono solamente quattro: EI, DI, LD A, I, LD I, A.
Pertanto, queste 4 istruzioni devono essere eseguite dal debugger, senza passare il controllo al programma utente, ma via software, eseguendo dei trasferimenti e delle operazioni di *set* e *reset* sulle locazioni della tavola dei registri che contengono i registri A, I e IFF.
Inoltre, come accennato in precedenza, due istruzioni non possono essere assolutamente eseguite in single step; queste sono IM 0 e IM 1, poiché programmano la CPU Z80 ad un funzionamento diverso dal modo vettorizzato di gestione degli interrupts (modo 2) e, d'altra parte, non possono essere eseguite via software poiché l'hardware della CPU Z80 non consente ad un programma di riconoscere in quale modo di interrupt la CPU si trova.

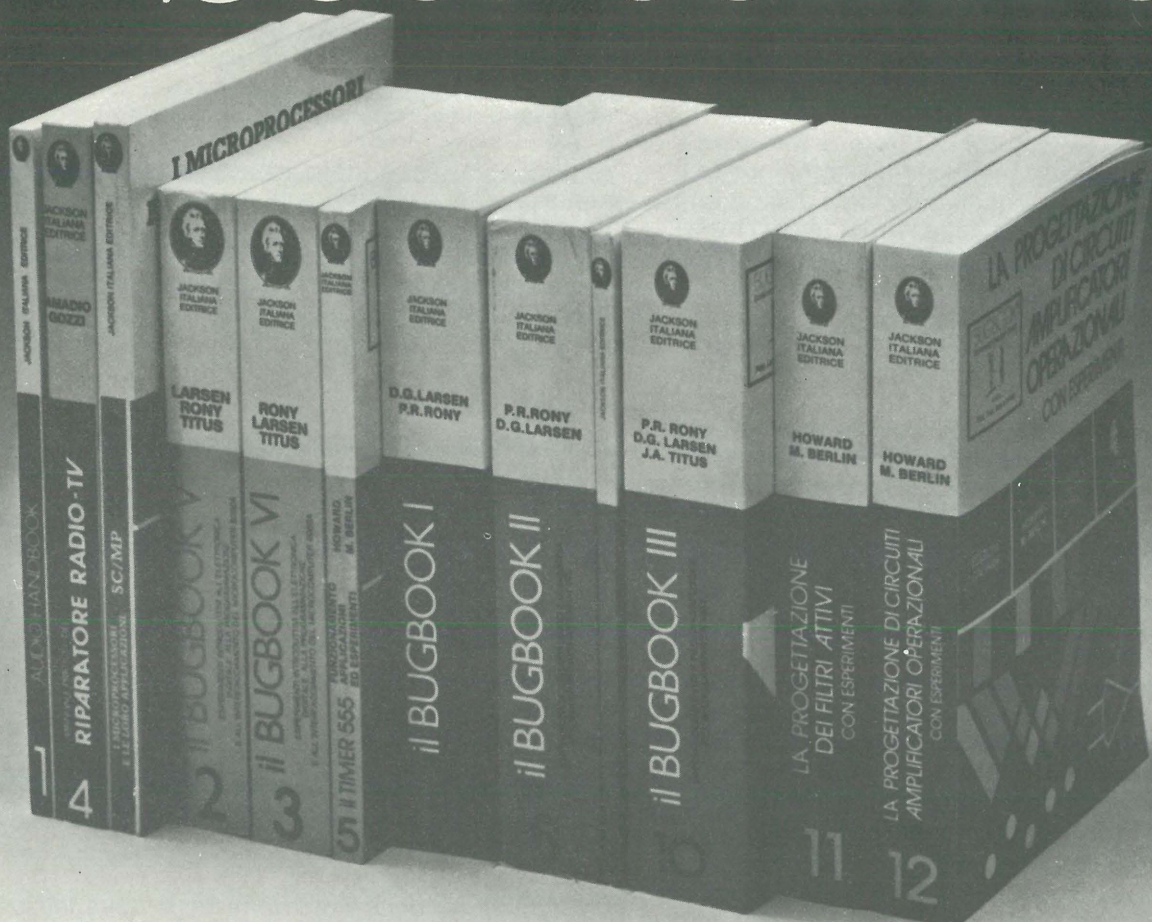
Figura 3 - Sequenza delle operazioni di debugger per il single step.

In dettaglio, la sequenza delle operazioni che il debugger esegue per implementare il *single step* sono:

- Esecuzione via software delle istruzioni EI, DI, LD A, I e LD, I, A se l'istruzione corrente è una di queste; altrimenti:
- Generazione di un interrupt locale sul PIO con la CPU avente interrupts disabilitati(*interrupt pending)*
- Salvataggio di una locazione di memoria *(buffer di single step)* del byte all'indirizzo PC utente - 1 e sua sostituzione con l'istruzione EI
- Ripristino dei registri contenuti nella tavola dei registri, tranne IFF che rimane a 0 e I che punta sulla routine SAVE
- Salto a *PC utente - 1*
- Esecuzione della istruzione EI e della istruzione corrente
- Rientro in debugger mediante interrupt vettorizzato
- Salvataggio nella tavola dei registri di tutti i registri della CPU tranne IFF e I che, non essendo stati stati modificati dalla istruzione corrente, rimangono eguali ai valori che erano stati memorizzati nella tavola dei registri e al verificarsi di un breakpoint o dall'esecuzione via software di una delle 4 istruzioni precedenti.
- Ripristino della locazione salvata nel buffer di single step.

La figura 3 illustra queste operazioni i nomi di registri della CPU Z80 seguiti dal nome USER indicano i registri residenti nella tavola dei registri.
La sostituzione del byte precedente l'istruzione corrente con EI è invisibile per l'utilizzatore, poiché il contenuto originale è salvato, prima dalla sostituzione, nel buffer di single step, e viene poi ripristinato dopo l'esecuzioine dell'istruzione utente.
L'interrupt di rientro è generato in un PIO del quale una porta è programmata in modo di controllo (modo 3) e che, normalmente, è utilizzato per l'interfaccia con una periferica del sistema.

Il debugger, prima di passare il controllo al programma utente, invia alla porta un dato particolare, in corrispondenza del quale il PIO genera una richiesta di interrupt che viene poi riconosciuta dalla CPU dopo l'istruzione successiva all'istruzione di EI.

■

i "best-sellers"

UDIO HANDBOOK

Jn manuale di progettazione audio con discussioni parti- colareggiate e progetti completi.

.. 9.500 (Abb. L. 8.550)

MANUALE PRATICO DEL RIPARATORE RADIO-TV

Jn autentico strumento di lavoro. Fra i numerosi argo- nenti trattati figurano: il laboratorio. Il servizio a domicilio. Antenne singole e centralizzate. Riparazione dei TV b/n e colore. Il ricevitore AM FM. Apparecchi e BF e CB. Stru- nentazione. Elenco ditte di radiotecnica, ecc.

.. 18.500 (Abb. L. 16.650)

SC/MP

Questo testo sul microprocessore SC/MP è corredato da una serie di esempi di applicazione e di programmi di uti- ità generale, tali da permettere al lettore una immediata verifica dei concetti teorici esposti e un'immediata speri- mentazione anche a livello di realizzazione progettuale.

L. 9.500 (Abb. L. 8.550)

IL BUGBOOK V E IL BUGBOOK VI

Esperimenti introduttivi all'elettronica digitale, alla pro- grammazione ed all'interfacciamento del microprocesso- re 8080A. I Bugbook V e VI costituiscono i primi veri testi organici a livello universitario sui microprocessori, con taglio nettamente sperimentale. Questi testi, oltre al Virgi- nia Polytechnic Institute, sono utilizzati in corsi aziendali, in seminari di aggiornamento tecnico e in scuole di tutto il mondo.

L. 19.000 ogni volume (Abb. L. 17.100)

IL TIMER 555

Il 555 è un temporizzatore dai mille usi. Il libro descrive circa 100 circuiti utilizzanti questo dispositivo e numerosi esperimenti.

L. 8.600 (Abb. L. 7.750)

IL BUGBOOK I E IL BUGBOOK II

Strumenti di studio per i neofiti e di aggiornamento pro- fessionale per chi già vive l'elettronica "tradizionale", questi due libri complementari presentano esperimenti sui circuiti logici e di memoria, utilizzanti circuiti integrati TTL. La teoria è subito collegata alla sperimentazione pra- tica, secondo il principio per cui si può veramente imparare solo quello che si sperimenta in prima persona.

L. 18.000 ogni volume (Abb. L. 16.200)

IL BUGBOOK II/A

Esperimenti di interfacciamento e trasmissione dati utiliz- zanti il ricevitore/trasmettitore universale asincrono (Uart) ed il Loop di corrente a 20 mA.

L. 4.500 (Abb. L. 4.050)

IL BUGBOOK III

Questo libro fornisce una parola definitiva sull'argomento "8080A" divenuto ormai un classico nella letteratura tecnica sui microprocessori. Da ogni parte, sia da istituti di formazione che da varie case costruttrici, sono stati pubblicati manuali e libri di testo, ma nessuno raggiunge la completezza di questo Bugbook e, soprattutto, nessuno presenta l'oggetto "8080A" in un modo così didattico e sperimentale.

L. 19.000 (Abb. L. 17.100)

LA PROGETTAZIONE DEI FILTRI ATTIVI CON ESPERIMENTI

Tratta un argomento di notevole attualità, rendendolo piano e comprensibile a tutti. Le riviste di settore dedica- no ampio spazio a questo aspetto dell'elettronica da oltre tre anni. Questo libro raccoglie tutto quanto è necessario sapere sui filtri attivi aggiungendovi numerosi esempi pratici ed esperimenti.

L. 15.000 (Abb. L. 13.500)

LA PROGETTAZIONE DEGLI AMPLIFICATORI OPERAZIONALI CON ESPERIMENTI

Gli amplificatori operazionali, in gergo chiamati OP-AMP, sono ormai diffusissimi in elettronica. Il libro ne spiega il funzionamento illustra alcune applicazioni pratiche e for- nisce numerosi esperimenti. Le persone interessate all'ar- gomento sono moltissime: dal tecnico esperto al semplice hobbista. Si tratta del miglior libro pubblicato nella mate- ria specifica.

L. 15.000 (Abb. L. 13.500)

Bit

CEDOLA DI COMMISSIONE LIBRARIA Da inviare a Jackson Italiana Editrice srl - Piazzale Massari, 22 - 20125 Milano.

SCONTO 10% AGLI ABBONATI

☐ Inviatemi i seguenti volumi pagherò al postino l'importo indicato più le spese di spedizione.

☐ Pagamento anticipato senza spese di spedizione.

Nome ____________

Cognome ____________

Via ____________ N. ______

Città ____________ Cap. ______

Codice Fiscale ____________

Data ____________ Firma ____________

N.	Titolo	Prezzo	Abb.
N. ——	Audio Handbook	**L. 9.500**	(Abb. L. 8.550)
N. ——	Manuale del Riparatore Radio-TV	**L. 18.500**	(Abb. L. 16.650)
N. ——	SC/MP	**L. 9.500**	(Abb. L. 8.550)
N. ——	Bugbook V	**L. 19.000**	(Abb. L. 17.100)
N. ——	Bugbook VI	**L. 19.000**	(Abb. L. 17.100)
N. ——	Timer 555	**L. 8.600**	(Abb. L. 7.750)
N. ——	Bugbook I	**L. 18.000**	(Abb. L. 16.200)
N. ——	Bugbook II	**L. 18.000**	(Abb. L. 16.200)
N. ——	Bugbook II/A	**L. 4.500**	(Abb. L. 4.050)
N. ——	Bugbook III	**L. 19.000**	(Abb. L. 17.100)
N. ——	La Progettazione dei Filtri Attivi	**L. 15.000**	(Abb. L. 13.500)
N. ——	La Progettazione degli Amp. Op.	**L. 15.000**	(Abb. L. 13.500)

Interprete Basic in 8080

Parte II. di **F. Maddaleno**

L'interprete Basic TBI può essere facilmente modificato ed esteso con l'aggiunta di altri comandi ed altre funzioni. I comandi possono essere di tipo diretto, cioè dati solo da tastiera per essere eseguiti immediatamente, oppure possono essere generali, ed essere dati da tastiera per l'esecuzione immediata o messi nel programma di utente per essere eseguiti durante l'interpretazione. Quando il TBI incontra un comando, cerca in una apposita tabella l'indirizzo della routine che lo esegue, e passa il controllo ad essa.
Volendo quindi cambiare od aggiungere altri comandi è sufficiente modificare la tabella e scrivere nuove routine. Un discorso analogo vale per le funzioni, anch'esse riconosciute per mezzo di una apposita tabella. Le tabelle e le routine di ricerca si trovano nella sezione di programma:

'TABELLE ... DIRECT ... EXEC'.

Esistono tabelle separate per i comandi, le funzioni, le parole TO e STEP e gli operatori di relazione. La tabella dei comandi è a sua volta divisa in due parti: la prima (TAB1) contiene i comandi di tipo diretto, la seconda (TAB2) contiene i comandi generali che possono essere richiamati indifferentemente da tastiera o da programma.
Tutte le tabelle (eccetto TAB1) terminano con un indirizzo di salto (due byte) a cui il TBI passa se non ha trovato la voce che stava cercando: tale indirizzo deve sempre essere l'ultima voce di ogni tabella.
Quando il TBI interpreta un comando dato da tastiera, comincia la ricerca in TAB1 con la routine DIRECT; quando invece il TBI incontra un comando in un programma, comincia la ricerca a partire da TAB2 con la routine EXEC. Questa routine non contiene l'indirizzo della tabella in cui cercare: il punto iniziale della ricerca deve essere fornito dalla routine che passa il controllo ad EXEC: ciò permette di usare EXEC non solo per la ricerca dei comandi, ma anche delle funzioni degli operatori di relazione ecc. Volendo ad esempio modificare la struttura dell'IF nella forma IF - THEN - ELSE, si possono aggiungere altre tabelle in modo analogo a quanto è fatto per il TO e lo STEP e si usa ancora per il riconoscimento la routine EXEC opportunamente inizializzata.
Le modifiche più semplici sono però i cambiamenti delle parole che identificano i comandi (es. RETURN) e l'aggiunta di altre voci cioè di altri comandi o funzioni. Modificare i nomi dei comandi è estremamente semplice: basta scrivere al posto del precedente il nuovo nome e ricompattare la tabella in cui si è fatta la modifica, se il nome nuovo non ha la stessa lunghezza di quello precedente: non bisogna infatti lasciare "buchi" all'interno della singola tabella e ovviamente neanche sovrapporre due voci!
Se si vuole ad esempio cambiare il nome RETURN in RET basta cambiare il codice a partire dalla locazione 150H fino alla 174H in questo modo:

```
150   524554          DEFM 'RET'
153   82              DEFB (RETURN+8000H). SHR.8
154   B9              DEFB RETURN&OFFH
155   52454D          DEFM 'REM'
158   83              DEFB (REM+8000H).SHR.8
...   ..              ....
173   84              DEFB (DEFLT+8000H).SHR.8
174   09              DEFB DEFLT&OFFH
...   ..              ....
...   ..        TAB4  ....
183   524E44          DEFM 'RND'
...   ..              ....
```

Conviene cioè allungare o accorciare lo spazio libero in fondo a TAB2 e TAB4 e non spostare l'indirizzo iniziale delle tabelle TAB4 TAB5 ecc. ad altri valori per non dover andare a modificare tutte le istruzioni nel TBI che fanno riferimento alle etichette spostate.
Se nell'esempio precedente fossero state spostate di tre byte anche le tabelle successive, si sarebbero poi dovute modificare tutte le istruzioni riferentisi alle etichette TAB4, TAB5 eccetera.
Per aggiungere invece nuovi comandi o nuove funzioni bisogna seguire questa procedura: si scrive la routine che realizza il nuovo comando o funzione (in fondo al TBI, prima che cominci la RAM ci sono circa 70 byte liberi). Si aggiunge poi nella tabella dei comandi o delle funzioni una nuova voce formata dal nome in codice ASCII, dal byte alto dell'indirizzo della routine col bit più significativo posto a uno, e dal byte meno significativo. La nuova voce può essere messa fra due altre voci qualunque già presenti in tabella, facendole spazio spostando parte della tabella e occupando i byte liberi che ci sono in fondo a TAB2 e a TAB4. Ovviamente per non dover spostare dei pezzi lunghi di tabella, conviene mettere la nuova voce verso la fine della tabella stessa.
La nuova voce non deve però essere messa dopo l'ultima voce, poichè in questo caso non sarebbe mai riconosciuta.
La tabella di comandi deve dunque sempre terminare così:

```
84      DEFB (DEFLT+8000H).SHR.8
09      DEFB DEFLT&OFFH
```

GAME TORINO

a cura del Gruppo Amatoriale Micro-Elaboratori, Torino

e analogamente la tabella delle funzioni deve sempre terminare con:

```
85        DEFB (XP40+8000H).SHR.8
08        DEFB XP40&OFFH
```

Queste voci senza stringa dicono dove saltare nel caso non venga trovata la parola che il TBI deve interpretare.
Supponiamo ad esempio di voler aggiungere la funzione radice quadrata con nome SQR e di scrivere la routine di nome RADICE, che la calcola, in fondo al TBI a partire dall'indirizzo 07B6H. Si aggiunge allora la stringa SQR in codice ASCII nella TAB4 (ad esempio dopo la funzione SIZE), si mettono poi il byte alto dell'indirizzo della routine RADICE con il bit 7 a uno (cioè 87H) e il byte basso dell'indirizzo (cioè B6H): la tabella va poi chiusa con i due byte che la chiudevano in precedenza. Si ottiene quindi:

```
18D    53495A45            DEFM 'SIZE'
191    85                  DEFB (SIZE+8000).SHR.8
192    63                  DEFB SIZE&OFFH
193    535152              DEFM 'SQR'
196    87                  DEFB (RADICE+8000).SHR.8
197    B6                  DEFB RADICE&OFFH
198    85                  DEFB (XP40+8000H).SHR.8
199    08                  DEFB XP40&OFFH
...    ..                  ....
...    ..          TAB5    ....
1A3    544F                DEFM 'TO'
```

Come in precedenza si sfrutta lo spazio libero in fondo alla TAB4 e non si sposta TAB5. Lo stesso procedimento vale per la TAB2: le nuove voci possono essere inserite dopo la voce STOP.
Se infine si vuole aggiungere un comando solo diretto (ad esempio SYS per tornare sotto sistema), questo deve essere messo in TAB1: non essendoci qui spazio libero bisogna anche spostare tutta la tabella seguente e quindi l'etichetta TAB2 che deve essere corretta in tutti i punti in cui compare nel programma (dalla cross reference table riportata in fondo al listato, si vede che essa compare solo nella linea 463).
In un prossimo articolo verrà descritta una estensione a questo TBI comprendente i comandi OUT, POKE e le funzioni IN, PEEK e SQR.

Nota per la lettura del listato

L'assemblatore, per motivi di spazio, in corrispondenza delle DEFM riporta nel listato (colonna OBJ) solo i primi 4 byte, mentre l'oggetto prodotto contiene ovviamente anche gli altri. Dovendo caricare l'oggetto letto dal listato, bisogna tenere conto di ciò: ad esempio alla riga 291, dove c'è DEFM 'RETURN'; l'oggetto da caricare è:

```
52 45 54 55 52 4E 82 B9 ...
R  E  T  U  R  N
```

L'assemblatore non stampa sul listato gli ultimi due byte della parola 52 e 4E corrispondenti a 'RN', ma ne tiene conto nel calcolo degli indirizzi e l'indirizzo di salto 82 B9 è correttamente posto nei byte 156H e 157H (vedere listato).
Un'ultima nota riguarda alcuni errori di stampa presenti nel testo dell Iª parte (BIT n.4): dove si parla della rilocazione dell'interpretatore (pag. 69) e più precisamente dei caratteri che indicano le linee di programma su cui intervenire, il carattere in bianco è @ . ■

```
                 869  ; <EXPR3> ::= <EXPR4> ! <EXPR3>*<EXPR4> ! <EXPR3>/<EXPR4>
                 870  ; <EXPR4> ::= <VAR> ! <FUNZIONE> ! (<EXPR>) ! <VETT> ! <NUM>
                 871  ; <VAR> ::= A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
                 872  ; <FUNZIONE> ::= ABS(<EXPR>) ! RND(<EXPR>) ! SIZE
                 873  ; <VETT> ::= @(<EXPR>)
                 874  ; <NUM> ::= <CIFRA> ! <NUM><CIFRA>
                 875  ; <CIFRA> ::= 1 2 3 4 5 6 7 8 9 0
                 876  ; IL SEGNO '!' SIGNIFICA 'OPPURE' . <EXPR> E' RECURSIVO
041D  CD6504   R 877  EXPR    CALL EXPR2
0420  E5         878          PUSH HL              ;SALVA IL VALORE DI <EXPR2>
0421  21B001   R 879  EXPR1   LD HL,TAB8-1         ;CERCA UN OPERATORE DI RELAZIONE
0424  C3CA01   R 880          JP EXEC              ;E LO ESEGUE
0427  CD5004   R 881  XP11    CALL XP18            ;OP. DI REL. '>='
042A  D8         882          RET C                ;RELAZIONE NON VERIFICATA  HL=0
042B  6F         883          LD L,A               ;RELAZIONE VERIFICATA  HL=1
042C  C9         884          RET
042D  CD5004   R 885  XP12    CALL XP18            ;OPERATORE DI RELAZIONE '#'
0430  C8         886          RET Z                ;FALSO : TORNA CON HL=0
0431  6F         887          LD L,A               ;VERO : METTE HL=1
0432  C9         888          RET
0433  CD5004   R 889  XP13    CALL XP18            ;OPERATORE DI RELAZIONE '>'
0436  C8         890          RET Z                ;FALSO
0437  D8         891          RET C                ;ANCORA FALSO : HL=0
0438  6F         892          LD L,A               ;VERO : HL=1
0439  C9         893          RET
043A  CD5004   R 894  XP14    CALL XP18            ;OPERATORE DI RELAZIONE '<='
043D  6F         895          LD L,A               ;PONE HL=1

                              TINY.BASIC      21$GIUGNO$1979      PAGE  17
 LOC  OBJ CODE M STMT SOURCE STATEMENT                             BAU 4.1

043E  C8         896          RET Z                ;SE SONO ATTIVI Z OPPURE C
043F  D8         897          RET C                ;LA RELAZIONE E' VERA E RITORNA
0440  6C         898          LD L,H               ;ALTRIMENTI RIAZZERA HL
0441  C9         899          RET
0442  CD5004   R 900  XP15    CALL XP18            ;OPERATORE DI RELAZIONE '='
0445  C0         901          RET NZ               ;FALSO : RITORNA CON HL=0
0446  6F         902          LD L,A               ;VERO : HL=1
0447  C9         903          RET
0448  CD5004   R 904  XP16    CALL XP18            ;OPERATORE DI RELAZIONE '<'
044B  DC         905          RET NC               ;FALSO : TORNA CON HL=0
044C  6F         906          LD L,A               ;ALTRIMENTI PONE HL=1
044D  C9         907          RET
044E  E1         908  XP17    POP HL               ;NON C'E' NESSUN OPERATORE DI REL.
044F  C9         909          RET                  ;TORNA COL VALORE DI <EXPR2> IN HL
0450  79         910  XP18    LD A,C               ;ROUTINE PER TUTTI GLI OP. DI REL.
0451  E1         911          POP HL               ;SCAMBIA I DUE ELEMENTI
0452  C1         912          POP BC               ;IN CIMA ALLO STACK
0453  E5         913          PUSH HL
0454  C5         914          PUSH BC
0455  4F         915          LD C,A
0456  CD6504   R 916          CALL EXPR2           ;CALCOLA LA SECONDA <EXPR2>
0459  EB         917          EX DE,HL             ;E METTE IL RISULTATO IN DE
045A  E3         918          EX (SP),HL           ;IN HL VALORE DELLA PRIMA <EXPR2>
045B  CD9C05   R 919          CALL CKHLDE          ;CONFRONTA PRIMA E SECONDA <EXPR2>
045E  D1         920          POP DE               ;RIPRISTINA IL PUNTATORE AL TESTO
045F  210000     921          LD HL,0              ;PONE HL=0 E A=1 PER COMODITA'
0462  3E01       922          LD A,1               ;DELLA ROUTINE CHIAMANTE
0464  C9         923          RET
                 924  ;
0465  CDC900   R 925  EXPR2   CALL TSTC            ;SEGNO NEGATIVO?
0468  2D         926          DEFB '-'
0469  06         927          DEFB (XP21-$-1)&0FFH
046A  210000     928          LD HL,0              ;SI',E' COME SE FOSSE '0-'
046D  C39704   R 929          JP XP26              ;LO TRATTA COME UNA SOTTRAZIONE
0470  CDC900   R 930  XP21    CALL TSTC            ;SEGNO POSITIVO?
0473  2B         931          DEFB '+'             ;LO IGNORA
0474  00         932          DEFB (XP22-$-1)&0FFH
0475  CDA104   R 933  XP22    CALL EXPR3           ;VALUTA LA PRIMA <EXPR3>
0478  CDC900   R 934  XP23    CALL TSTC            ;C'E' UNA ADDIZIONE?
047B  2B         935          DEFB '+'
047C  15         936          DEFB (XP25-$-1)&0FFH
047D  E5         937          PUSH HL              ;SI':SALVA IL VALORE
047E  CDA104   R 938          CALL EXPR3           ;VALUTA LA SECONDA <EXPR3>
0481  EB         939  XP24    EX DE,HL             ;SECONDA <EXPR3> IN DE
0482  E3         940          EX (SP),HL           ;PRIMA <EXPR3> IN HL
0483  7C         941          LD A,H               ;CONFRONTA IL SEGNO
0484  AA         942          XOR D
0485  7A         943          LD A,D
```

```
0486  19            944                ADD HL,DE
0487  D1            945                POP DE                 ;RIPRISTINA IL PUNTATORE AL TESTO
0488  FA7804   R    946                JP M,XP23              ;LE 2 <EXPR3> HANNO SEGNO DIVERSO
048B  AC            947                XOR H                  ;LE DUE <EXPR3> HANNO STESSO SEGNO
048C  F27804   R    948                JP P,XP23              ;COME IL RISULTATO
048F  C30701   R    949                JP QHOW                ;ALTRIMENTI C'E' STATO OVERFLOW
0492  CDC900   R    950  XP25          CALL TSTC              ;C'E' UNA SOTTRAZIONE?
0495  2D            951                DEFB '-'
0496  8F            952                DEFB (XP42-$-1)&0FFH
0497  E5            953  XP26          PUSH HL                ;SI':SALVA LA PRIMA <EXPR3>

                     TINY.BASIC      21$GIUGNO$1979      PAGE  18
LOC   OBJ CODE M STMT SOURCE STATEMENT                    BAU 4.1

0498  CDA104   R    954                CALL EXPR3             ;CALCOLA LA SECONDA <EXPR3>
049B  CD9005   R    955                CALL CHGSGN            ;LE CAMBIA SEGNO
049E  C38104   R    956                JP XP24                ;E LA SOMMA
                    957  ;
04A1  CD0205   R    958  EXPR3         CALL EXPR4             ;VALUTA LA PRIMA <EXPR4>
04A4  CDC900   R    959  XP31          CALL TSTC              ;C'E' UNA MOLTIPLICAZIONE?
04A7  2A            960                DEFB '*'
04A8  2C            961                DEFB (XP34-$-1)&0FFH
04A9  E5            962                PUSH HL                ;SI':SALVA LA PRIMA <EXPR4>
04AA  CD0205   R    963                CALL EXPR4             ;CALCOLA LA SECONDA <EXPR4>
04AD  0600          964                LD B,0                 ;AZZERA B PER IL SEGNO
04AF  CD8D05   R    965                CALL CHKSGN            ;CONTROLLA IL SEGNO DI HL
04B2  EB            966                EX DE,HL               ;SECONDA <EXPR4> IN DE
04B3  E3            967                EX (SP),HL             ;PRIMA <EXPR4> IN HL
04B4  CD8D05   R    968                CALL CHKSGN            ;CONTROLLA IL SEGNO DI HL
04B7  7C            969                LD A,H                 ;HL > 255?
04B8  B7            970                OR A
04B9  CAC204   R    971                JP Z,XP32              ;NO
04BC  7A            972                LD A,D                 ;SI', E COME E' DE?
04BD  B2            973                OR D
04BE  EB            974                EX DE,HL               ;METTE IL PIU' PICCOLO IN HL
04BF  C20801   R    975                JP NZ,AHOW             ;ANCHE DE > 255 : OVERFLOW!
04C2  7D            976  XP32          LD A,L
04C3  210000        977                LD HL,0                ;AZZERA IL RISULTATO
04C6  B7            978                OR A                   ;MOLTIPLICA FACENDO SOLO
04C7  CAF404   R    979                JP Z,XP35              ;SOMME E CONTANDOLE CON A
04CA  19            980  XP33          ADD HL,DE
04CB  DA0801   R    981                JP C,AHOW              ;SE C'E' CARRY,C'E' OVERFLOW
04CE  3D            982                DEC A
04CF  C2CA04   R    983                JP NZ,XP33
04D2  C3F404   R    984                JP XP35                ;FINITO
04D5  CDC900   R    985  XP34          CALL TSTC              ;C'E' UNA DIVISIONE?
04D8  2F            986                DEFB '/'
04D9  4C            987                DEFB (XP42-$-1)&0FFH
04DA  E5            988                PUSH HL                ;SALVA LA PRIMA <EXPR4>
04DB  CD0205   R    989                CALL EXPR4             ;E VALUTA LA SECONDA <EXPR4>
04DE  0600          990                LD B,0                 ;AZZERA B PER IL SEGNO
04E0  CD8D05   R    991                CALL CHKSGN            ;CONTROLLA SEGNO SEC. <EXPR4>
04E3  EB            992                EX DE,HL               ;SEC. <EXPR4> IN DE
04E4  E3            993                EX (SP),HL             ;PRIMA <EXPR4> IN HL
04E5  CD8D05   R    994                CALL CHKSGN            ;CONTROLLA IL SEGNO DI HL
04E8  7A            995                LD A,D                 ;DIVIDE PER 0?
04E9  B3            996                OR E
04EA  CA0801   R    997                JP Z,AHOW              ;SI':SEGNALA L'ERRORE
04ED  C5            998                PUSH BC                ;SALVA IL SEGNO DEL RISULTATO
04EE  CD7005   R    999                CALL DIVIDE            ;FA ESEGUIRE LA DIVISIONE
04F1  60           1000                LD H,B                 ;METTE IL RISULTATO IN HL
04F2  69           1001                LD L,C
04F3  C1           1002                POP BC                 ;RECUPERA IL SEGNO DEL RISULTATO
04F4  D1           1003  XP35          POP DE                 ;RIPRISTINA IL PUNTATORE AL TESTO
04F5  7C           1004                LD A,H                 ;HL DEVE ESSERE POSITIVO
04F6  B7           1005                OR A
04F7  FA0701   R   1006                JP M,QHOW              ;SE NO,C'E' STATO OVERFLOW
04FA  78           1007                LD A,B                 ;SEGNO DEL RISULTATO NEGATIVO?
04FB  B7           1008                OR A
04FC  FC9005   R   1009                CALL M,CHGSGN          ;CAMBIA SEGNO SE NECESSARIO
04FF  C3A404   R   1010                JP XP31                ;CERCA SE CI SONO ALTRI TERMINI
                   1011

                     TINY.BASIC      21$GIUGNO$1979      PAGE  19
LOC   OBJ CODE M STMT SOURCE STATEMENT                    BAU 4.1

0502  218201   R   1012  EXPR4         LD HL,TAB4-1           ;CERCA LA FUNZIONE IN TAB4
0505  C3CA01   R   1013                JP EXEC                ;E LA ESEGUE
0508  CD9500   R   1014  XP40          CALL TSTV              ;NO, NON E' UNA FUNZIONE
050B  DA1305   R   1015                JP C,XP41              ;NEPPURE UNA VARIABILE
050E  7E           1016                LD A,(HL)              ;E' UNA VARIABILE
050F  23           1017                INC HL
0510  66           1018                LD H,(HL)              ;NE METTE IL VALORE IN HL
0511  6F           1019                LD L,A
0512  C9           1020                RET
0513  CDDD00   R   1021  XP41          CALL TSTNUM            ;OPPURE E' UN NUMERO
0516  78           1022                LD A,B                 ;IN B C'E' IL NUMERO DI CIFRE
0517  B7           1023                OR A
0518  C0           1024                RET NZ                 ;E' PROPRIO UN NUMERO
0519  CDC900   R   1025  PARN          CALL TSTC              ;NON CI SONO CIFRE

00D3  4E            177                LD C,(HL)              ;VECCHIO PC, FA CIOE' UN SALTO
0448  CD5004   R    904  XP16          CALL XP18              ;OPERATORE DI RELAZIONE '<'
051C  28           1026                DEFB '('               ;ALLORA DEVE ESSERE '(<EXPR>)'
051D  09           1027                DEFB (XP43-$-1)&0FFH
051E  CD1D04   R   1028                CALL EXPR              ;VALUTA (<EXPR>)
0521  CDC900   R   1029                CALL TSTC              ;E' CHIUSA LA PARENTESI?
0524  29           1030                DEFB ')'
0525  01           1031                DEFB (XP43-$-1)&0FFH
0526  C9           1032  XP42          RET
0527  C3D805   R   1033  XP43          JP QWHAT               ;NON E' CHIUSA:SEGNALA L'ERRORE
                   1034  ;
052A  CD1905   R   1035  RND           CALL PARN              ;FUNZIONE   ' RND (<EXPR>) '
052D  7C           1036                LD A,H                 ;(<EXPR>) DEVE ESSERE POSITIVA
052E  B7           1037                OR A
052F  FA0701   R   1038                JP M,QHOW
0532  B5           1039                OR L
0533  CA0701   R   1040                JP Z,QHOW              ;E DIVERSA DA ZERO
0536  D5           1041                PUSH DE                ;SALVA (<EXPR>) E IL PUNTATORE
0537  E5           1042                PUSH HL
0538  2A1108   R   1043                LD HL,(RANPNT)         ;LEGGE LA MEM. IN UN PUNTO A CASO
053B  11FF07   R   1044                LD DE,LSTROM           ;CONTROLLA DI NON ESSERE FUORI
053E  CD8000   R   1045                CALL COMP
0541  DA4705   R   1046                JP C,RA1               ;NON E' FUORI
0544  210000   R   1047                LD HL,START
0547  5E           1048  RA1           LD E,(HL)
0548  23           1049                INC HL
0549  56           1050                LD D,(HL)
054A  221108   R   1051                LD (RANPNT),HL
054D  E1           1052                POP HL
054E  EB           1053                EX DE,HL
054F  C5           1054                PUSH BC
0550  CD7005   R   1055                CALL DIVIDE            ;RND (N) = MOD (M,N) + 1
0553  C1           1056                POP BC
0554  D1           1057                POP DE
0555  23           1058                INC HL
0556  C9           1059                RET
                   1060  ;
0557  CD1905   R   1061  ABS           CALL PARN              ;FUNZIONE   ' ABS (<EXPR>) '
055A  CD8D05   R   1062                CALL CHKSGN            ;CONTROLLA IL SEGNO
055D  7C           1063                LD A,H                 ;NOTARE CHE -32768
055E  B4           1064                OR H                   ;NON PUO' CAMBIARE SEGNO
055F  FA0701   R   1065                JP M,QHOW              ;SEGNALA L'ERRORE
0562  C9           1066                RET
0563  2A1308   R   1067  SIZE          LD HL,(TXTUNF)         ;FUNZIONE   ' SIZE '
0566  D5           1068                PUSH DE                ;FORNISCE IL NUMERO DI
0567  EB           1069                EX DE,HL               ;BYTE LIBERI TRA 'TXTUNF'

                     TINY.BASIC      21$GIUGNO$1979      PAGE  20
LOC   OBJ CODE M STMT SOURCE STATEMENT                    BAU 4.1

0568  21001F   @   1070                LD HL,VARBGN           ;E 'VARBGN'
056B  CD8605   R   1071                CALL SUBDE
056E  D1           1072                POP DE
056F  C9           1073                RET
                   1074  ;                                                                        ;
                   1075  ;                                                                        ;
                   1076  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   1077  ;                                                                        ;
                   1078  ;                                                                        ;
                   1079  ;.........DIVIDE.........SUBDE.......CHKSGN......CHGSGN.......CKHLDE..........;
                   1080  ;                                                                        ;
                   1081  ;                                                                        ;
                   1082  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   1083  ;                                                                        ;
                   1084  ;                                                                        ;
                   1085  ; 'DIVIDE' DIVIDE HL PER DE, RISULTATO IN BC, RESTO IN HL                ;
                   1086  ;                                                                        ;
                   1087  ; 'SUBDE' SOTTRAE DA HL IL CONTENUTO DI DE                               ;
                   1088  ;                                                                        ;
                   1089  ; 'CHKSGN' CONTROLLA IL SEGNO DI HL. SE E' POSITIVO NON FA NULLA E RITORNA, ;
                   1090  ; SE E' NEGATIVO CAMBIA IL SEGNO DI HL E INVERTE IL BIT 7 DEL REGISTRO  B. ;
                   1091  ;                                                                        ;
                   1092  ; 'CHGSGN' CAMBIA INCONDIZIONATAMENTE IL SEGNO DI HL E COMMUTA IL BIT 7 DI B. ;
                   1093  ;                                                                        ;
                   1094  ; 'CKHLDE' CONTROLLA IL SEGNO DI HL E DI DE. SE SONO DIVERSI SCAMBIA HL CON DE ;
                   1095  ; SE SONO UGUALI NON LI SCAMBIA. IN ENTRAMBI I CASI CONFRONTA HL CON DE E ;
                   1096  ; POSIZIONA CORRETTAMENTE I FLAG.                                       ;
                   1097  ;                                                                        ;
                   1098  ;                                                                        ;
0570  E5           1099  DIVIDE        PUSH HL
0571  6C           1100                LD L,H                 ;DIVIDE H PER DE
0572  2600         1101                LD H,0
0574  CD7B05   R   1102                CALL DV1
0577  41           1103                LD B,C                 ;SALVA IL RISULTATO IN B
0578  7D           1104                LD A,L                 ;(RESTO+L) DIVISO DE
0579  E1           1105                POP HL
057A  67           1106                LD H,A
057B  0EFF         1107  DV1           LD C,-1                ;RISULTATO IN C
057D  0C           1108  DV2           INC C
057E  CD8605   R   1109                CALL SUBDE             ;DIVIDE CON SOTTRAZIONI
```

```
0581  D27D05    R 1110            JP NC,DV2        ;SUCCESSIVE CONTANDOLE CON C
0584  19          1111            ADD HL,DE
0585  C9          1112            RET
                  1113  ;
0586  7D          1114  SUBDE     LD A,L           ;ESEGUE L'OPERAZIONE HL=HL-DE
0587  93          1115            SUB E
0588  6F          1116            LD L,A
0589  7C          1117            LD A,H
058A  9A          1118            SBC A,D
058B  67          1119            LD H,A
058C  C9          1120            RET
                  1121  ;
058D  7C          1122  CHKSGN    LD A,H           ;CONTROLLA IL SEGNO DI HL
058E  B7          1123            OR A
058F  F0          1124            RET P            ;SE NEGATIVO LO CAMBIA
                  1125  ;
0590  7C          1126  CHGSGN    LD A,H           ;CAMBIA IL SEGNO DI HL
0591  2F          1127            CPL

                      TINY.BASIC      21$GIUGNO$1979      PAGE  21
 LOC  OBJ CODE M STMT SOURCE STATEMENT                     BAU 4.1

0592  67          1128            LD H,A
0593  7D          1129            LD A,L
0594  2F          1130            CPL
0595  6F          1131            LD L,A
0596  23          1132            INC HL
0597  78          1133            LD A,B
0598  EE80        1134            XOR 80H          ;INVERTE LO STATO DEL BIT 7 DI B
059A  47          1135            LD B,A
059B  C9          1136            RET
                  1137  ;
059C  7C          1138  CKHLDE    LD A,H           ;CONFRONTO DEI SEGNI DI HL E DE
059D  AA          1139            XOR D            ;STESSO SEGNO?
059E  F2A205    R 1140            JP P,CK1         ;SI': CONFRONTA DE CON HL
05A1  EB          1141            EX DE,HL         ;NO,SCAMBIA DE CON HL E CONFRONTA
05A2  CD8000    R 1142  CK1       CALL COMP
05A5  C9          1143            RET
                  1144  ;                                                                        ;
                  1145  ;                                                                        ;
                  1146  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  1147  ;                                                                        ;
                  1148  ;                                                                        ;
                  1149  ;........SETVAL........FIN........ENDCHK........ERROR.(ET.ALII)................;
                  1268  ; E' PRESENTE UNA LINEA CON UN NUMERO MAGGIORE, DE PUNTA A QUEST'ULTIMA ED I ;
                  1269  ; FLAG SONO POSIZIONATI NC E NZ. INFINE SE ARRIVA AL TERMINE DELLA SAVEAREA  ;
                  1270  ; SENZA AVER TROVATO NE' LA LINEA CERCATA NE' UNA CON NUMERO MAGGIORE,TORNA ;
                  1271  ; CON I FLAG POSIZIONATI NZ E C.                                             ;
                  1272  ; 'FNDLN' INIZIALIZZA DE ALL'INIZIO DELLA SAVEAREA DEL PROGRAMMA D'UTENTE PER ;
                  1273  ; INIZIARE LA RICERCA. CI SONO ALCUNI ENTRY POINT CHE NON INIZIALIZZANO DE.  ;
                  1274  ; 'FNDLNP' COMINCIA LA RICERCA DI UNA LINEA CON NUMERO DATO A PARTIRE DA DE  ;
                  1275  ; 'FNDNXT' INCREMENTA DE DI 2, CERCA UN CR E INIZIA LA RICERCA DELLA LINEA   ;
                  1276  ; CON IL NUMERO DATO.                                                        ;
                  1277  ; 'FNDSKP' CERCA UN CR E POI INIZIA LA RICERCA DELLA LINEA CON NUMERO DATO.  ;
                  1278  ;                                                                        ;
                  1279  ;                                                                        ;
060E  CD7707    R 1280  GETLN     CALL OUTC        ;STAMPA IL CAR. DI PRESENTAZIONE
0611  11371F    @ 1281            LD DE,BUFFER
0614  CD9707    R 1282  GL1       CALL CHKIO       ;INTERROGA LA TASTIERA
0617  CA1406    R 1283            JP Z,GL1         ;NESSUN CARATTERE : ASPETTA
061A  CD7707    R 1284            CALL OUTC        ;C'E' UN CARATTERE : FA L'ECO
061D  FE0A        1285            CP 0AH           ;IGNORA I LINE-FEED
061F  CA1406    R 1286            JP Z,GL1
0622  B7          1287            OR A             ;IGNORA I NULL
0623  CA1406    R 1288            JP Z,GL1
0626  FE08        1289            CP 08H           ;E' UN BACKSPACE? (CONTROL-H)
0628  CA3B06    R 1290            JP Z,GL3         ;VA A CANCELLARE L'ULTIMO CAR.
062B  FE7F        1291            CP 7FH           ;E' UN RUBOUT?
062D  CA4F06    R 1292            JP Z,GL4         ;VA A CANCELLARE TUTTA LA LINEA
0630  12          1293            LD (DE),A        ;ALTRIMENTI SALVA IL CARATTERE
0631  13          1294            INC DE           ;E AGGIORNA IL PUNTATORE
0632  FE0D        1295            CP CR            ;IL CAR. ERA UN CARRIAGE-RETURN?
0634  C8          1296            RET Z            ;SI',E' FINITA LA LINEA
0635  7B          1297            LD A,E           ;ALTRIMENTI C'E' ANCORA SPAZIO?
0636  FE7F      X 1298            CP BUFEND&0FFH   ;CONFRONTA CON IND.BASSO FINE BUFFER
0638  C21406    R 1299            JP NZ,GL1        ;SI',ASPETTA UN ALTRO CARATTERE
063B  7B          1300  GL3       LD A,E           ;CANCELLA L'ULTIMO CARATTERE
063C  FE37      X 1301            CP BUFFER&0FFH   ;MA C'ERA ALMENO UN CARATTERE?

                      TINY.BASIC      21$GIUGNO$1979      PAGE  24
 LOC  OBJ CODE M STMT SOURCE STATEMENT                     BAU 4.1

063E  CA4F06    R 1302            JP Z,GL4         ;NO,LEGGE UNA LINEA INTERA
0641  1B          1303            DEC DE           ;SI',TIRA INDIETRO IL PUNTATORE
0642  3E20        1304            LD A,' '         ;FA USCIRE UNO SPAZIO
0644  CD7707    R 1305            CALL OUTC        ;PER TOGLIERE IL CARATTERE ERRATO
0647  3E08        1306            LD A,08H
0649  CD7707    R 1307            CALL OUTC
```

```
0694  3E22        1393            LD A,'"'         ;C'ERA UN '"' :
0696  CD8106    R 1394  QT1       CALL PRTSTG      ;STAMPA FINO AL SUCCESSIVO '"'
0699  FE0D        1395            CP CR            ;L'ULTIMO CARATTERE ERA UN CR?
069B  E1          1396            POP HL           ;RECUPERA L'INDIRIZZO DI RITORNO
069C  CA0F02    R 1397            JP Z,RUNNXL      ;ERA UN CR : ESEGUE LA LINEA DOPO
069F  23          1398  QT2       INC HL           ;SALTA 3 BYTE AL RITORNO
06A0  23          1399            INC HL           ;NELLA ROUTINE CHIAMANTE
06A1  23          1400            INC HL
06A2  E9          1401            JP (HL)
06A3  CDC900    R 1402  QT3       CALL TSTC        ;C'E' UN ' ?
06A6  27          1403            DEFB ''''

                      TINY.BASIC      21$GIUGNO$1979      PAGE  26
 LOC  OBJ CODE M STMT SOURCE STATEMENT                     BAU 4.1

06A7  05          1404            DEFB (QT4-$-1)&0FFH
06A8  3E27        1405            LD A,''''        ;SI COMPORTA COME CON "
06AA  C39606    R 1406            JP QT1
06AD  CDC900    R 1407  QT4       CALL TSTC        ;C'E' UN BACK-ARROW? ( _ )
06B0  5F          1408            DEFB '_'         ;(CIOE' FRECCIA ALL'INDIETRO <- ?)
06B1  0C          1409            DEFB (QT5-$-1)&0FFH
06B2  3E8D        1410            LD A,8DH         ;SI':STAMPA CR SENZA LF
06B4  CD7707    R 1411            CALL OUTC        ;PER NON FAR RICONOSCERE CR A OUTC
06B7  CD7707    R 1412            CALL OUTC        ;METTE BIT 7 A 1 ; LO FA USCIRE
                  1413                             ;DUE VOLTE PER DAR TEMPO ALLA TTY
06BA  E1          1414            POP HL           ;RECUPERA L'INDIRIZZO DI RITORNO
06BB  C39F06    R 1415            JP QT2
06BE  C9          1416  QT5       RET              ;NON HA TROVATO ' O " O BACK-ARROW
                  1417  ;
06BF  D5          1418  PRTNUM    PUSH DE          ;SALVA DE
06C0  110A00      1419            LD DE,10D        ;BASE DIECI
06C3  D5          1420            PUSH DE
06C4  42          1421            LD B,D           ;B=SEGNO
06C5  0D          1422            DEC C            ;C=SPAZI
06C6  CD8D05    R 1423            CALL CHKSGN      ;CONTROLLA IL SEGNO
06C9  F2CF06    R 1424            JP P,PN1         ;SEGNO POSITIVO:NON LO STAMPA
06CC  062D        1425            LD B,'-'         ;B CONTIENE IL SEGNO
06CE  0D          1426            DEC C            ;VIENE CONTATO LO SPAZIO PER IL -
06CF  C5          1427  PN1       PUSH BC          ;SALVA SEGNO E NUMERO SPAZI
06D0  CD7005    R 1428  PN2       CALL DIVIDE      ;DIVIDE HL PER 10
06D3  78          1429            LD A,B           ;RISULTATO UGUALE A ZERO?
06D4  B1          1430            OR C
06D5  CAE006    R 1431            JP Z,PN3         ;SI',LA CONVERSIONE E' FINITA
06D8  E3          1432            EX (SP),HL       ;NO,SALVA IL RESTO DELLA DIVISIONE
06D9  2D          1433            DEC L            ;CONTA GLI SPAZI DA STAMPARE
06DA  E5          1434            PUSH HL          ;HL E' IL VECCHIO BC
06DB  60          1435            LD H,B           ;SPOSTA IL RISULTATO
06DC  69          1436            LD L,C
06DD  C3D006    R 1437            JP PN2           ;E LO RIDIVIDE PER 10
06E0  C1          1438  PN3       POP BC           ;TUTTE LE CIFRE DELLA CONVERSIONE
06E1  0D          1439  PN4       DEC C            ;SONO NELLO STACK
06E2  79          1440            LD A,C           ;CI SONO SPAZI DA STAMPARE?
06E3  B7          1441            OR A
06E4  FAEF06    R 1442            JP M,PN5         ;NO
06E7  3E20        1443            LD A,' '         ;SI'
06E9  CD7707    R 1444            CALL OUTC        ;LI STAMPA
06EC  C3E106    R 1445            JP PN4           ;CE NE SONO ALTRI DA STAMPARE?
06EF  78          1446  PN5       LD A,B           ;STAMPA IL SEGNO (NULL O -)
06F0  CD7707    R 1447            CALL OUTC
06F3  5D          1448            LD E,L           ;METTE L'ULTIMO RESTO IN E
06F4  7B          1449  PN6       LD A,E           ;CONTROLLA LA CIFRA IN E
06F5  FE0A        1450            CP 10D           ;10 SIGNIFICA FINITO
06F7  D1          1451            POP DE
06F8  C8          1452            RET Z            ;SE E'COST' RITORNA
06F9  C630        1453            ADD A,'0'        ;CONVERTE LA CIFRA IN ASCII
06FB  CD7707    R 1454            CALL OUTC        ;E LA STAMPA
06FE  C3F406    R 1455            JP PN6           ;TORNA A STAMPARE LA SUCCESSIVA
                  1456  ;
0701  1A          1457  PRTLN     LD A,(DE)        ;CARICA LA PARTE BASSA
0702  6F          1458            LD L,A           ;DEL NUMERO DELLA LINEA IN HL
0703  13          1459            INC DE
0704  1A          1460            LD A,(DE)        ;IDEM PER LA PARTE ALTA
0705  67          1461            LD H,A

                      TINY.BASIC      21$GIUGNO$1979      PAGE  27
 LOC  OBJ CODE M STMT SOURCE STATEMENT                     BAU 4.1

0706  13          1462            INC DE
0707  0E04        1463            LD C,04H         ;STAMPA 4 CIFRE
0709  CDBF06    R 1464            CALL PRTNUM      ;DEL NUMERO DELLA LINEA
070C  3E20        1465            LD A,' '         ;CON UNO SPAZIO DIETRO
070E  CD7707    R 1466            CALL OUTC
0711  97          1467            SUB A
0712  CD8106    R 1468            CALL PRTSTG      ;E IL TESTO DI SEGUITO
0715  C9          1469            RET
                  1470  ;
                  1471  ;
```

```
064C  C31406   R 1308           JP GL1            ;VA A LEGGERE ALTRI CARATTERI
064F  CD7507   R 1309  GL4      CALL CRLF         ;LEGGE UNA LINEA INTERA
0652  3E5E       1310           LD A,'^'          ;FA USCIRE CR-LF E UP-ARROW
0654  C30E06   R 1311           JP GETLN          ;LEGGE UNA LINEA
                 1312  ;
0657  7C         1313  FNDLN    LD A,H            ;CONTROLLA IL SEGNO DI HL
0658  B7         1314           OR A
0659  FA0701   R 1315           JP M,QHOW         ;NON PUO' ESSERE NEGATIVO
065C  111508   R 1316           LD DE,TXTBGN      ;INIZIALIZZA IL PUNTATORE AL TESTO
                 1317  ;
                 1318  FNDLNP
065F  E5         1319  FL1      PUSH HL           ;SALVA IL NUMERO DELLA LINEA
0660  2A1308   R 1320           LD HL,(TXTUNF)    ;PROVA SE HA OLTREPASSATO LA FINE
0663  2B         1321           DEC HL
0664  CD8000   R 1322           CALL COMP
0667  E1         1323           POP HL            ;RECUPERA IL NUMERO DELLA LINEA
0668  D8         1324           RET C             ;C,NZ : OLTREPASSATA LA FINE
0669  1A         1325           LD A,(DE)         ;FINE TESTO NON SUPERATA :LEGGE
066A  95         1326           SUB L             ;IL PRIMO BYTE
066B  47         1327           LD B,A            ;E LO CONFRONTA
066C  13         1328           INC DE
066D  1A         1329           LD A,(DE)         ;LEGGE IL SECONDO BYTE
066E  9C         1330           SBC A,H           ;CONFRONTA IL BYTE ALTO DEL NUMERO
066F  DA7606   R 1331           JP C,FL2          ;NON E' ANCORA LA LINEA CERCATA
0672  1B         1332           DEC DE            ;ALTRIMENTI O E' TROVATA
0673  B0         1333           OR B              ;OPPURE NON C'E' IN MEMORIA
0674  C9         1334           RET               ;NC,Z=TROVATA ; NC,NZ=NON C'E'
                 1335  ;
                 1336  FNDNXT
0675  13         1337           INC DE            ;CERCA LA LINEA SUCCESSIVA
0676  13         1338  FL2      INC DE            ;SUBITO DOPO I PRIMI 2 BYTE
                 1339  ;
0677  1A         1340  FNDSKP   LD A,(DE)         ;CERCA UN CARRIAGE-RETURN
0678  FE0D       1341           CP CR
067A  C27606   R 1342           JP NZ,FL2         ;CONTINUA A CERCARE
067D  13         1343           INC DE            ;TROVATO : LO OLTREPASSA
067E  C35F06   R 1344           JP FL1            ;CONTINUA A CERCARE LA LINEA
                 1345                             ;CON NUMERO DATO IN HL
```

```
                         TINY.BASIC     21$GIUGNO$1979      PAGE  25
LOC  OBJ CODE M STMT SOURCE STATEMENT                        BAU 4.1

                 1346  *E
                 1347  ;                                                                          ;
                 1348  ;                                                                          ;
                 1349  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 1350  ;                                                                          ;
                 1351  ;                                                                          ;
                 1352  ;...........PRTSTG...........QTSTG...........PRTNUM..........PRTLN.............;
                 1353  ;                                                                          ;
                 1354  ;                                                                          ;
                 1355  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 1356  ;                                                                          ;
                 1357  ;                                                                          ;
                 1358  ; 'PRTSTG' STAMPA UNA STRINGA PUNTATA DA DE. TORNA AL CHIAMANTE QUANDO INCON- ;
                 1359  ; TRA NEL TESTO UN CR OPPURE QUANDO IL BYTE CHE STA PER STAMPARE E' UGUALE A ;
                 1360  ; QUELLO CONTENUTO NEL REGISTRO A (FORNITO DAL CHIAMANTE). IL PRECEDENTE A E' ;
                 1361  ; SALVATO IN B E B VIENE PERSO.                                               ;
                 1362  ;                                                                          ;
                 1363  ; 'QTSTG' CERCA UN BACK-ARROW UN APICE O LE VIRGOLETTE. SE NON TROVA NESSUNO ;
                 1364  ; DI QUESTI CARATTERI TORNA AL CHIAMANTE. SE TROVA UN BACK-ARROW FA USCIRE UN ;
                 1365  ; CR SENZA LF. SE TROVA UN APICE O LE VIRGOLETTE (' O "), STAMPA LA STRINGA ;
                 1366  ; DOPO LA STAMPA RITORNA AL CHIAMANTE SALTANDO PERO' I 3 BYTE SUCCESSIVI ALLA ;
                 1367  ; CALL (CHE SONO NORMALMENTE UNA ISTRUZIONE DI SALTO)                         ;
                 1368  ;                                                                          ;
                 1369  ; 'PRTNUM' STAMPA IL NUMERO IN HL. SE NECESSARIO VENGONO AGGIUNTI DEI BLANK ;
                 1370  ; NON SIGNIFICATIVI IN MODO DA UTILIZZARE IL NUMERO DI POSIZIONI DI STAMPA  ;
                 1371  ; SPECIFICATE IN C. SE IL NUMERO DI CIFRE DA STAMPARE SUPERA IL VALORE DI C, ;
                 1372  ; IL NUMERO VIENE STAMPATO TUTTO UGUALMENTE. VIENE EVENTUALMENTE ANCHE STAM- ;
                 1373  ; PATO IL SEGNO NEGATIVO E CONTEGGIATO NEL NUMERO DI POSIZIONI DI STAMPA.    ;
                 1374  ; IL SEGNO POSITIVO NON VIENE MAI STAMPATO.                                  ;
                 1375  ;                                                                          ;
                 1376  ; 'PRTLN' STAMPA UNA LINEA COMPLETA (COMPRESO IL NUMERO) PRENDENDOLA DALLA   ;
                 1377  ; SAVEAREA.                                                                 ;
                 1378  ;                                                                          ;
                 1379  ;
0681  47         1380  PRTSTG   LD B,A            ;SALVA IL CARATTERE DI CONTROLLO
0682  1A         1381  PS1      LD A,(DE)         ;PRENDE UN CARATTERE DA STAMPARE
0683  13         1382           INC DE            ;AGGIORNA IL PUNTATORE AL TESTO
0684  B8         1383           CP B              ;E' UGUALE AL CAR. DI CONTROLLO?
0685  C8         1384           RET Z             ;SI' : RITORNA
0686  CD7707   R 1385           CALL OUTC         ;ALTRIMENTI LO STAMPA
0689  FE0D       1386           CP CR             ;ERA UN CARRIAGE-RETURN?
068B  C28206   R 1387           JP NZ,PS1         ;NO,CONTINUA
068E  C9         1388           RET               ;SI',RITORNA
                 1389  ;
068F  CDC900   R 1390  QTSTG    CALL TSTC
0692  22         1391           DEFB '"'
0693  0F         1392           DEFB (QT3-$-1)&0FFH
```

```
                 1472  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 1473  ;                                                                          ;
                 1474  ;                                                                          ;
                 1475  ;............MVUP...........MVDOWN.............POPA.........PUSHA..............;
                 1476  ;                                                                          ;
                 1477  ;                                                                          ;
                 1478  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 1479  ;                                                                          ;
                 1480  ;                                                                          ;
                 1481  ; 'MVUP' SPOSTA IN SU UN BLOCCO DA DOVE -> DE A DOVE -> BC FINCHE  DE=HL     ;
                 1482  ;                                                                          ;
                 1483  ; 'MVDOWN' SPOSTA IN GIU' UN BLOCCO DA DOVE ->DE A DOVE ->HL FINCHE'  DE=BC  ;
                 1484  ;                                                                          ;
                 1485  ; 'POPA' RIPRISTINA LA SAVEAREA DELLE VARIABILI DEI LOOP (FATTI COI 'FOR'),  ;
                 1486  ; PRENDENDO I VALORI PRECEDENTI DALLO STACK.                                 ;
                 1487  ;                                                                          ;
                 1488  ; 'PUSHA' SALVA LA SAVEAREA DELLE VARIABILI DEI LOOP NELLO STACK.            ;
                 1489  ;                                                                          ;
                 1490  ;                                                                          ;
0716  CD8000   R 1491  MVUP     CALL COMP
0719  C8         1492           RET Z             ;SE HL=DE HA FINITO E RITORNA
071A  1A         1493           LD A,(DE)         ;PRENDE UN BYTE
071B  02         1494           LD (BC),A         ;E LO TRASFERISCE
071C  13         1495           INC DE
071D  03         1496           INC BC
071E  C31607   R 1497           JP MVUP           ;FINCHE' HA FINITO
                 1498  ;
0721  78         1499  MVDOWN   LD A,B
0722  92         1500           SUB D             ;CONTROLLA SE DE=BC
0723  C22907   R 1501           JP NZ,MD1         ;CONTINUA IL TRASFERIMENTO
0726  79         1502           LD A,C            ;B=D : FORSE HA FINITO
0727  93         1503           SUB E             ;CONTROLLA ANCHE SE C=E
0728  C8         1504           RET Z             ;SI',HA PROPRIO FINITO E TORNA
0729  1B         1505  MD1      DEC DE            ;ALTRIMENTI TRASFERISCE UN BYTE
072A  2B         1506           DEC HL            ;PERO' PRIMA DECREMENTA
072B  1A         1507           LD A,(DE)         ;ENTRAMBI I PUNTATORI
072C  77         1508           LD (HL),A
072D  C32107   R 1509           JP MVDOWN         ;E TORNA DA CAPO
                 1510  ;
0730  C1         1511  POPA     POP BC            ;IN BC C'E' L'IND. DI RITORNO
0731  E1         1512           POP HL            ;RIPRISTINA LOPVAR
0732  220708   R 1513           LD (LOPVAR),HL    ;MA LOPVAR=0 SIGNIFICA CHE
0735  7C         1514           LD A,H            ;NON CI SONO PIU' LOPVAR
0736  B5         1515           OR L              ;CIOE' C'E' UNA POPA DI TROPPO
0737  CA4A07   R 1516           JP Z,PP1          ;RITORNA CON HL=0
073A  E1         1517           POP HL            ;HL NON E' ZERO
073B  220908   R 1518           LD (LOPINC),HL    ;RIPRISTINA LE ALTRE VARIABILI
073E  E1         1519           POP HL
```

```
                         TINY.BASIC     21$GIUGNO$1979      PAGE  28
LOC  OBJ CODE M STMT SOURCE STATEMENT                        BAU 4.1

073F  220B08   R 1520           LD (LOPLMT),HL
0742  E1         1521           POP HL
0743  220D08   R 1522           LD (LOPLN),HL
0746  E1         1523           POP HL
0747  220F08   R 1524           LD (LOPPT),HL
074A  C5         1525  PP1      PUSH BC
074B  C9         1526           RET
                 1527  ;
074C  21A71F   @ 1528  PUSHA    LD HL,STKLMT
074F  CD9005   R 1529           CALL CHGSGN
0752  C1         1530           POP BC            ;IN BC INDIRIZZO DI RITORNO
0753  39         1531           ADD HL,SP         ;LO STACK E' QUASI PIENO?
0754  D20706   R 1532           JP NC,QSORRY      ;SI':PECCATO ,STAMPA 'SORRY'
0757  2A0708   R 1533           LD HL,(LOPVAR)    ;NO,SALVA LE VARIABILI DEL LOOP
075A  7C         1534           LD A,H            ;MA SE LOPVAR=0
075B  B5         1535           OR L              ;SALVA SOLO LOPVAR
075C  CA7207   R 1536           JP Z,PU1          ;CHE FUNZIONA DA DELIMITATORE
075F  2A0F08   R 1537           LD HL,(LOPPT)     ;ALTRIMENTI SALVA ANCHE
0762  E5         1538           PUSH HL           ;TUTTE LE ALTRE
0763  2A0D08   R 1539           LD HL,(LOPLN)
0766  E5         1540           PUSH HL
0767  2A0B08   R 1541           LD HL,(LOPLMT)
076A  E5         1542           PUSH HL
076B  2A0908   R 1543           LD HL,(LOPINC)
076E  E5         1544           PUSH HL
076F  2A0708   R 1545           LD HL,(LOPVAR)
0772  E5         1546  PU1      PUSH HL
0773  C5         1547           PUSH BC           ;BC CONTIENE L'IND. DI RITORNO
0774  C9         1548           RET
                 1549  ;                                                                          ;
                 1550  ;                                                                          ;
                 1551  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 1552  ;                                                                          ;
                 1553  ;                                                                          ;
                 1554  ;...........CRLF..............OUTC..............CHKIO.........................;
                 1555  ;                                                                          ;
                 1556  ;                                                                          ;
```

```
                1557 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                1558 ;                                                                          ;
                1559 ;                                                                          ;
                1560 ; 'CRLF' FA USCIRE UN CARRIAGE-RETURN E UN LINE-FEED SULLA STAMPANTE.      ;
                1561 ;                                                                          ;
                1562 ; 'OUTC' E' CONTROLLATA DA UNA ABILITAZIONE SOFTWARE (VARIABILE OCSW).     ;
                1563 ; SE OCSW VALE ZERO, OUTC TORNA AL CHIAMANTE SENZA STAMPARE NULLA. SE INVECE ;
                1564 ; OCSW E' DIVERSA DA ZERO, FA USCIRE IL BYTE PRESENTE IN A. SE QUESTO E' UN ;
                1565 ; CARRIAGE-RETURN, LA ROUTINE AGGIUNGE AUTOMATICAMENTE ANCHE UN LINE-FEED. ;
                1566 ; LA ROUTINE ALTERA SOLO I FLAG.                                           ;
                1567 ;                                                                          ;
                1568 ; 'CHKIO' CONTROLLA L'INGRESSO. SE NON CI SONO CARATTERI, RITORNA CON IL FLAG ;
                1569 ; DI ZERO ATTIVO. SE CI SONO CARATTERI IN INGRESSO, RESETTA IL FLAG ZERO E ;
                1570 ; TORNA COL CARATTERE IN A. PERO' SE IL CARATTERE E' UN CONTROL-O, CAMBIA LO ;
                1571 ; STATO DI OCSW (CIOE' ABILITA O DISABILITA LA STAMPA), E VIENE ATTIVATO IL ;
                1572 ; FLAG DI ZERO. SE IL CARATTERE E' UN CONTROL-C, LA ROUTINE SALTA DIRETTAMENTE ;
                1573 ; A START, E QUINDI L'INTERPRETATORE SI RIPRESENTA DA CAPO.                ;
                1574 ;                                                                          ;
                1575 ;                                                                          ;
0775  3E0D      1576 CRLF    LD A,CR          ;PREPARA IN A UN CARRIAGE-RETURN
                1577 ;

          TINY.BASIC        21$GIUGNO$1979      PAGE  29
LOC   OBJ CODE M STMT SOURCE STATEMENT                BAU 4.1

0777  F5          1578 OUTC    PUSH AF          ;STAMPA UN CARATTERE
0778  3A0008    R 1579         LD A,(OCSW)      ;SOLO SE LO SWITCH OCSW
077B  B7          1580         OR A             ;E' ATTIVO
077C  C28207    R 1581 OC2     JP NZ,OC3        ;E' ATTIVO : STAMPA
077F  F1          1582         POP AF           ;E' SPENTO :RIPRISTINA A
0780  00          1583         NOP
0781  C9          1584         RET              ;E RITORNA
0782  DBDF        1585 OC3     IN A,(UARST)     ;SI PREPARA A STAMPARE
0784  E601        1586         AND FLAOUT       ;CONTROLLA IL BIT DI STATUS
0786  CA8207    R 1587         JP Z,OC3         ;NON E' PRONTO : ASPETTA
0789  F1          1588         POP AF           ;PRONTO :
078A  D3DE        1589         OUT (UARDAT),A   ;FA USCIRE IL DATO
078C  FE0D        1590         CP CR            ;ERA UN CARRIAGE-RETURN?
078E  C0          1591         RET NZ           ;NO , RITORNA
078F  3E0A        1592         LD A,0AH         ;SI', AGGIUNGE UN LINE-FEED
0791  CD7707    R 1593         CALL OUTC        ;LA ROUTINE E' RECURSIVA
0794  3E0D        1594         LD A,CR          ;RIPRISTINA IN A IL CR
0796  C9          1595         RET
0797  DBDF        1596 CHKIO   IN A,(UARST)     ;C'E' UN DATO PRONTO?
0799  00          1597         NOP
079A  E602        1598         AND FLAGIN
079C  C8          1599         RET Z            ;NESSUN DATO,TORNA CON A=0
079D  DBDE        1600         IN A,(UARDAT)    ;LEGGE IL DATO
079F  E67F        1601         AND 7FH          ;MASCHERA IL BIT 7
07A1  FE0F        1602         CP 0FH           ;E' UN CONTROL-O ?
07A3  C2B007    R 1603         JP NZ,CI1        ;NO , CONTINUA I TEST
07A6  3A0008    R 1604         LD A,(OCSW)      ;SI', INVERTE LO STATO
07A9  2F          1605         CPL              ;DELLO SWITCH OCSW
07AA  320008    R 1606         LD (OCSW),A
07AD  C39707    R 1607         JP CHKIO         ;TORNA AD ASPETTARE UN ALTRO CAR.
07B0  FE03        1608 CI1     CP 03H           ;E' UN CONTROL-C ?
07B2  C0          1609         RET NZ           ;NO , RITORNA
07B3  C30000    R 1610         JP START         ;SI', RIPARTE DA CAPO
                  1611 ;
                  1612 ;
                  1613 LSTROM  EQU START+7FFH   ;DEFINIZIONE PARAMETRI
                  1614 CR      EQU   0DH        ;DEFINIZIONE DEL CARRIAGE-RETURN
                  1615 UARST   EQU   0DFH       ;INDIR. STATO UART
                  1616 UARDAT  EQU   0DEH       ;INDIR. DATO UART
                  1617 FLAGIN  EQU   02H        ;MASCHERA BIT DATO DISPONIBILE
                  1618 FLAOUT  EQU   01H        ;MASCHERA BIT TRASMET. VUOTO
0800              1619         ORG   START+800H
                  1620 ;
                  1621 ;       DI QUI I AVANTI CI DEVE ESSERE RAM
                  1622 ;
0800  FF          1623 OCSW    DEFB 0FFH        ;SWITCH DI STAMPA
0801  0000        1624 CURRNT  DEFW 0           ;PUNTA ALLA LINEA CORRENTE
0803  0000        1625 STKGOS  DEFW 0           ;SALVATAGGIO SP NELLE 'GOSUB'
                  1626 VARNXT  EQU $
0805  0000        1627 STKINP  DEFW 0           ;SALVATAGGIO SP NELLE 'INPUT'
0807  0000        1628 LOPVAR  DEFW 0           ;BUFFER SALVATAGGIO PER I LOOP
0809  0000        1629 LOPINC  DEFW 0           ;PASSO DI INCREMENTO
080B  0000        1630 LOPLMT  DEFW 0           ;VALORE FINALE
080D  0000        1631 LOPLN   DEFW 0           ;NUMERO LINEA
080F  0000        1632 LOPPT   DEFW 0           ;PUNTATORE AL TESTO
0811  0000        1633 RANPNT  DEFW START       ;PUNTATORE A NUMERO CASUALE
0813  1508        1634 TXTUNF  DEFW TXTBGN      ;PUNTATORE AREA ANCORA VUOTA
0815              1635 TXTBGN  DEFS 1           ;INIZIO AREA PROG. UTENTE

          TINY.BASIC        21$GIUGNO$1979      PAGE  30
LOC   OBJ CODE M STMT SOURCE STATEMENT                BAU 4.1
```

```
INPERR 03A5    784 1232
INPUT  03AF    791  301  302
IP1    03AF    792  814  839
IP2    03BF    798  794
IP3    03D1    810  797
IP4    03FD    835  796
IP5    0406    840  838
LET    040F    846  280  281  850
LIST   0239    510  267  268
LOPINC 0809   1629  654  709 1518 1543
LOPLMT 080B   1630  648  717 1520 1541
LOPLN  080D   1631  656  725 1522 1539
LOPPT  080F   1632  658  687  727 1524 1537
LOPVAR 0807   1628   51  589  644  660  696  713 1513 1533 1545
LS1    0242    513  517
LSTROM 07FF   1613 1044
LT1    041A    851  844  849
MD1    0729   1505 1501
MVDOWN 0721   1499  103  685 1509
MVUP   0716   1491   82  107 1497
NEW    01FA    442  273  274
NEXT   0339    691  277  278
NX0    0342    694  705
NX1    0374    722  720
NX2    0388    730  724
NX3    035C    706  701
OC2    077C   1581
OC3    0782   1585 1581 1587
OCSW   0800   1623 1579 1604 1606
OK     0114    223   45
OUTC   0777   1578 1238 1280 1284 1305 1307 1385 1411 1412 1444 1447
                   1454 1466 1593
PARN   0519   1025  146 1035 1061
PN1    06CF   1427 1424
PN2    06D0   1428 1437
PN3    06E0   1438 1431
PN4    06E1   1439 1445
PN5    06EF   1446 1442
PN6    06F4   1449 1455
POPA   0730   1511  604  703  730
PP1    074A   1525 1516
PR0    0269    530  527  542
PR1    0275    536  532
PR2    025E    525  522
PR3    027B    538  535  549
PR6    0286    543  540
PR8    028C    545  537
PRINT  0251    519  304  305
PRTLN  0701   1457  514 1233

CROSS REFERENCE          TINY.BASIC        21$GIUGNO$1979      PAGE  33
SYMBOL  VAL M DEFN REFS

PRTNUM 06BF   1418  547 1464
PRTSTG 0681   1380   47  806 1217 1240 1394 1468
PS1    0682   1381 1387
PU1    0772   1546 1536
PUSHA  074C   1528  579  641
QHOW   0107    218  148  196  949 1006 1038 1040 1065 1315
QSORRY 0607   1242  100 1532
QT1    0696   1394 1406
QT2    069F   1398 1415
QT3    06A3   1402 1392
QT4    06AD   1407 1404
QT5    06BE   1416 1409
QTSTG  068F   1390  536  793
QWHAT  05D8   1214  139  332  333  597  692  800 1033 1183 1196
RA1    0547   1048 1046
RANPNT 0811   1633 1043 1051
REM    038E    773  295  296
RETURN 02B9    593  292  293
RND    052A   1035  316  317
RUN    0209    449  270  271
RUNNXL 020F    452  529 1207 1397
RUNSML 021F    462  524  779 1252
RUNTSL 0218    456  474  592  781
SETVAL 05A6   1182  642  846
SIZE   0563   1067  151  322  323
SORRY  011E    227 1243
ST1    0007     44
ST2    0017     50   48
ST3    0020     53  108
ST4    0057     87   76
STACK  2000   1645   40
START  0000     39   92  447  454  513  782 1047 1229 1241 1610 1613
                   1619 1633 1636 1645
STKGOS 0803   1625   52  586  591  594  600
```

```
1F00            1636           ORG START+1F00H
1F00            1637  VARBGN   DEFS 54           ;VARIABILE @(0) E ALTRE VAR.
1F36            1638           DEFS 1            ;BYTE EXTRA PER IL BUFFER
1F37            1639  BUFFER   DEFS 72           ;BUFFER DI INGRESSO
                1640  BUFEND   EQU $             ;FINE BUFFER INGRESSO
                1641  TXTEND   EQU VARBGN        ;FINE AREA PROGRAMMA UTENTE
                1642                             ;E' QUI PER MOTIVI DI ASSEMBLATORE
1F7F            1643           DEFS 40           ;BYTE EXTRA PER LO STACK
                1644  STKLMT   EQU $             ;LIMITE SUPERIORE STACK
                1645  STACK    EQU START+2000H   ;PUNTO INIZIALE STACK
                1646           END

CROSS REFERENCE                  TINY.BASIC      21$GIUGNO$1979      PAGE  31
SYMBOL   VAL M DEFN REFS

ABS     0557    1061  319  320
AHOW    0108     219  471  583  975  981  997
ASORRY  0608    1243  153
AWHAT   05D9    1215  699
BUFEND  1F7F    1640 1298
BUFFER  1F37    1639   56  822 1281 1301
CHGSGN  0590    1126  955 1009 1529
CHKIO   0797    1596  462  515 1282 1607
CHKSGN  058D    1122  965  968  991  994 1062 1423
CI1     07B0    1608 1603
CK1     05A2    1142 1140
CKHLDE  059C    1138  722  919
COMP    0080     124   99  152  700 1045 1142 1322 1491
CR      000D    1614  222  224  226  228  526  843 1204 1211 1295 1341
                     1386 1395 1576 1590 1594
CRLF    0775    1576   44  523  528  543 1309
CURRNT  0801    1624   49  457  584  602  655  726  787  812  815  833
                     1223
DEFLT   0409     842  309  310
DIRECT  01C7     365   62
DIVIDE  0570    1099  999 1055 1428
DV1     057B    1107 1102
DV2     057D    1108 1110
ENDCHK  05D2    1210  442  446  449  469  511  593
ERROR   05DC    1216  220 1244
EX0     01CA     368  386
EX1     01CE     370  376
EX2     01E1     381  383
EX3     01EB     387  373
EX4     01ED     389  391
EX5     01F2     392  380
EXEC    01CA     367  464  646  650  880 1013
EXPR    041D     877  466  533  545  580  647  651  776  823 1028 1188
EXPR1   0421     879
EXPR2   0465     925  877  916
EXPR3   04A1     958  933  938  954
EXPR4   0502    1012  958  963  989
FI1     05C8    1203 1200
FI2     05D1    1208 1205
FIN     05BF    1198  138  541
FINISH  008E     137  544  605  689  729  731  840  851
FL1     065F    1319 1344
FL2     0676    1338 1331 1342
FLAGIN  0002    1617 1598
FLAOUT  0001    1618 1586
FNDLN   0657    1313   74  470  512  582
FNDLNP  065F    1318  453  516
FNDNXT  0675    1336   78
FNDSKP  0677    1340  780
FOR     02D4     641  298  299
FR1     02E4     647  330  331
FR2     02F0     651  336  337
FR3     02F6     653  338  339
FR4     02F9     654  652
FR5     02FC     655
FR7     0311     666  674  677
FR8     0332     687  670
GETLN   060E    1280   54  821 1311


GL3     063B    1300 1290
GL4     064F    1309 1292 1302
GOSUB   0297     579  289  290
GOTO    0228     466  286  287
HOW     010E     221  219
IF      0394     776  283  284
IF1     0397     777  774
IGNBLK  0086     131   58  135  141  172  189  368 1210
```

```
STKINP  0805    1627  784  818
STKLMT  1FA7    1644 1528
STOP    0203     446  307  308
SUBDE   0586    1114  155 1071 1109
SV1     05BC    1196 1187
TAB1    0124     265  365
TAB2    0134     275  463
TAB4    0183     314 1012
TAB5    01A3     328  645
TAB6    01A9     334  649
TAB8    01B1     341  879
TC1     00CE     174
TC2     00D9     182  175
TN1     00E4     190  215
TSTC    00C9     171  520  525  530  538  836  847  925  930  934  950
                      959  985 1025 1029 1185 1198 1203 1390 1402 1407
TSTNUM  00DD     187   57  510 1021
TSTV    0095     141  691  795  799 1014 1182
TV1     00B9     158  144
TXTBGN  0815    1635  443  450 1316 1634
TXTEND  1F00    1641   98
TXTUNF  0813    1634   81   86   88  101  444 1067 1320
UARDAT  00DE    1616 1589 1600
UARST   00DF    1615 1585 1596

CROSS REFERENCE                  TINY.BASIC      21$GIUGNO$1979      PAGE  34
SYMBOL   VAL M DEFN REFS

VARBGN  1F00    1637  154  162 1070 1641
VARNXT  0805    1626  693  704
WHAT    0117     225 1215
XP11    0427     881  343  344
XP12    042D     885  346  347
XP13    0433     889  349  350
XP14    043A     894  355  356
XP15    0442     900  352  353
XP16    0448     904  358  359
XP17    044E     908  360  361
XP18    0450     910  881  885  889  894  900  904
XP21    0470     930  927
XP22    0475     933  932
XP23    0478     934  946  948
XP24    0481     939  956
XP25    0492     950  936
XP26    0497     953  929
XP31    04A4     959 1010
XP32    04C2     976  971
XP33    04CA     980  983
XP34    04D5     985  961
XP35    04F4    1003  979  984
XP40    0508    1014  324  325
XP41    0513    1021 1015
XP42    0526    1032  952  987
XP43    0527    1033 1027 1031
```

LA NOTA

Riflessioni sui personal computers dopo lo SMAU

di **Aldo Cavalcoli**

Nell'ambito dell'ultimo SMAU di Milano, vi è stata un'ampia ed interessante rassegna di sistemi gestionali, personal computer orientati o orientabili all'office automation, presentati non solo da case, quali NIXDORF, OLIVETTI, IBM, SAGA, SAICO, ma anche da organizzazioni specializzate in personal computer e non solo in quelli, quali EDICONSULT, HOMIC, UNICOMP HARDEN etc.

Prima riflessione: il personal è presente all'EDP USA, a BIT 79, allo SMAU, al prossimo MICROBIAS, sostanzialmente è presente in tutti quei contesti in cui viene trattato l'argomento gestione delle informazioni.

E fin qui niente di nuovo, in quanto il personal computer è un computer e come tale realizza la gestione delle informazioni. La novità, se vogliamo definirla tale, sta nel fatto che il personal, non realizzato per una specifica applicazione, si pone come concorrente o alternativa a sistemi realizzati per specifiche applicazioni, quali appunto i sistemi gestionali presentati allo SMAU.

Dire concorrenza forse è eccessivo, meglio alternativa, rivolta verso quella fascia di utenti che non vedono nell'offerta mini una risposta soddisfacente soprattutto economicamente.

Detto questo, due miti da sfatare.

Il primo, quello per cui un personal costa meno di un gestionale chiavi in mano.

Infatti, dato un obiettivo, è nostra ferma convinzione che il suo raggiungimento abbia un costo, indipendente dai mezzi. Se da un lato l'obiettivo è raggiunto subito pagando moneta sonante, dall'altro si può pagare altrimenti con tempi più lunghi, impegno personale, necessità di autoistruzione.

Si noti bene: a parità di obiettivo, perchè c'è da sottolineare che i personal soddisfano anche, e solo loro, tutte quelle esigenze che escludono, per semplicità e valore, l'uso dei mini.

Secondo mito: una macchina più potente è preferibile ad una che lo è meno.

Questo non è vero nella misura in cui una grande potenza può essere inadeguata, se vogliamo sprecata, in molti casi.

Resta quindi valida l'equazione costi-tempi, intesi sia come tempo per raggiungere un dato obiettivo che come tempo da investire lavorativamente per permettere il raggiungimento di quell'obiettivo.

Le considerazioni precedenti sono nate da tutta una serie di colloqui avuti con potenziali utenti nell'ambito dello SMAU, oltre che in precedenti occasioni. Una schematizzazione di queste considerazioni può essere:

1) il personal computer costa poco ma non è corredato come mi servirebbe, d'altra parte un mini corredato costa troppo per la mia applicazione, e

2) nel personal, per un utilizzo efficiente, devo "coinvolgermi" personalmente.

Considerazione finale del potenziale utente: forse non è ancora maturo il momento, meglio aspettare.

Nostra conclusione: ciò indica che il mercato non è maturo non tanto come esigenze da soddisfare, ma come preparazione di base dell'utente.

Dato che non è corretto chiedere una preparazione di base dell'utente per il successo di un'iniziativa, occorre spianargli la strada.

A questo punto interviene l'elemento software, inteso come software applicativo molto diversificato, a basso costo, possibilmente orientato a categorie professionali.

La presenza di questo prodotto implica la necessità di altri investimenti, questa volta da parte dei venditori di personal.

Sarà seguita questa strada?

Combinatore telefonico con microprocessore

di **R. Dal Monte, A. Meneguzzo** - *Istituto Tecnico Statale a Ordinamento Speciale, Milano*

Premessa

Partendo da un'esigenza concreta della vita quotidiana, si è realizzata un'applicazione a microprocessore che, con chiare limitazioni di applicabilità pratica, ha una finalità puramente didattica, in quanto è stata attuata in ambito scolastico, servendosi di un sistema MMD1.
Proprio per il carattere didattico dell'esperienza, si è ritenuto opportuno corredare la presentazione con notizie e descrizioni utilizzabili come spunti per uno studio più approfondito, che potrà precedere o seguire una eventuale realizzazione pratica.

Presentazione del problema

In molte aziende il telefono è uno strumento di lavoro insostituibile. Ma può diventare stressante comporre durante la giornata decine di numeri telefonici, per di più col rischio di sbagliare. Per ovviare a questi inconvenienti, sono stati immessi sul mercato degli apparecchi che compongono automaticamente il numero telefonico desiderato.
Gli apparecchi più semplici si avvalgono di schede magnetiche (oppure perforate) sulle quali è scritto, in codice, il numero telefonico.
Il combinatore che abbiamo messo a punto utilizza

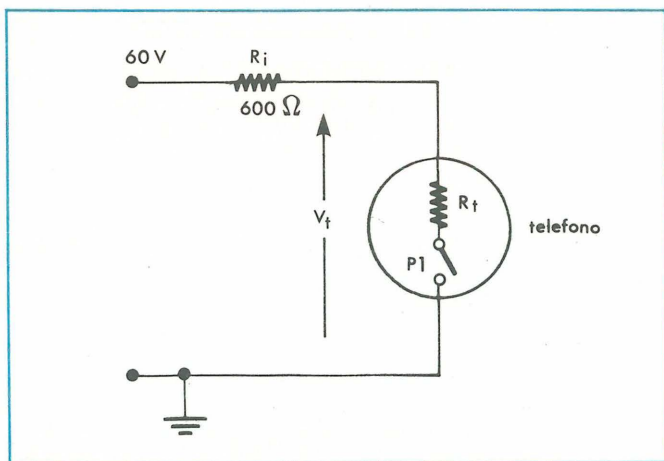



Figura 1 - Circuito telefonico.

un microcomputer MMD1, al quale è stata aggiunta una opportuna interfaccia.

Funzionamento del telefono

Il telefono è collegato alla rete telefonica mediante un cavetto bipolare. Tra i due conduttori vi è una tensione di circa 60 volt continui. La resistenza interna della linea telefonica si aggira sui 600 ohm. Il circuito è rappresentato schematicamente in Figura 1.
Quando il microtelefono è appeso, il pulsante P1 risulta aperto, e di conseguenza ai capi dell'apparecchio è presente una tensione di 60 volt. Quando viene sollevato il microtelefono, l'interruttore P1 si chiude e, poiché la resistenza interna del telefono è di circa 100 ohm, la tensione ai capi di quest'ultimo scende a 8.5 V. A questo punto, appena ricevuto dalla centrale il segnale di libero, è possibile comporre il numero desiderato.
Quando, per mezzo del disco combinatore, si invia sulla linea una cifra del numero da chiamare, al momento del rilascio l'interruttore P1 si apre e si chiude ritmicamente per un numero di volte che è funzione della cifra composta (componendo il numero 2, ad esempio, P1 si apre e si chiude due volte).
In Figura 2 è rappresentato il segnale telefonico quando, dopo che la centrale ha dato il segnale di libero, è composto il numero 2, e successivamente il numero 1. Le durate degli impulsi (tempi di apertura e di chiusura di P1) hanno tempi ben precisi. I valori

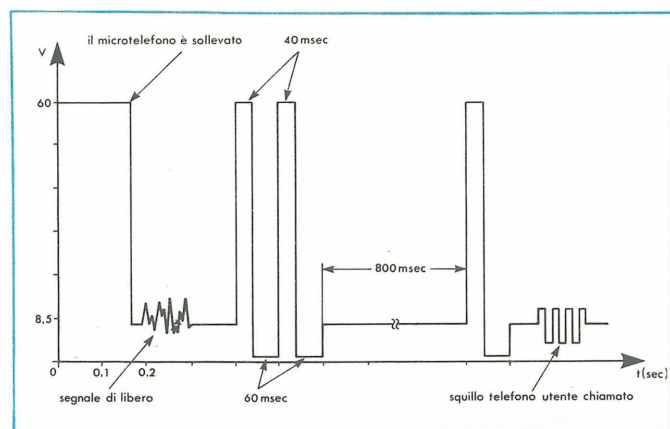



Figura 2 - Segnale telefonico durante la composizione di un numero.

standard di 40 msec per il tempo di apertura e di 60 msec per il tempo di chiusura. Inoltre tra una cifra e la successiva devono trascorrere almeno 800 msec.
Per quanto riguarda la parte fonica, questa é schematizzata in Figura 3. Il microfono, che permette di comunicare con l'utente chiamato, è a carbone, ed inserisce un segnale fonico che si somma alla corrente di linea. La voce dell'interlocutore giunge invece tramite una capsula magnetica (R). Il condensatore C ha la funzione di bloccare la componente continua del segnale.
Nello schema in figura é presente un secondo interruttore (P2), che ha la funzione di escludere la parte fonica del telefono durante la trasmissione del numero telefonico: infatti P2 si chiude non appena si ruota il disco combinatore. Annullando inoltre la resistenza interna del telefono, P2 fa in modo che durante la formazione del numero la tensione ai capi del telefono possa scendere, in regime impulsivo, a zero volt.

Ed ora al lavoro

Dato che è meglio non mettere le mani dentro il telefono, vediamo come, operando a livello di linea, si possa simulare la funzione del disco combinatore appena descritto.
In A, C ed E di Figura 4 si possono vedere i tre possibili stati del telefono, mentre in B, D ed F come sono realizzati con un dispositivo esterno collocato tra linea e apparecchio telefonico. PD e PR sono due pulsanti aggiuntivi inseriti nel dispositivo. AP è un amplificatore telefonico che si inserisce in linea

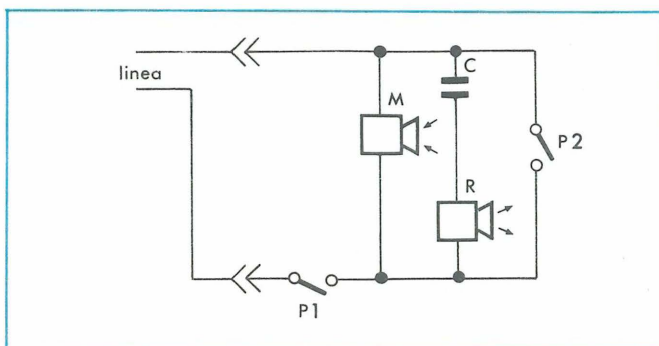



Figura 3 - Sezione fonica del telefono.

attraverso il contatto di riposo PR da una parte, e attraverso PI dall'altra.

- In A l'apparecchio telefonico è a riposo. E' abilitato il funzionamento della suoneria, ma non quello della parte fonica (essendo P1 aperto).
- In B la configurazione dei pulsanti PD, PR e PI non altera funzionalmente lo schema elettrico originario dell'apparecchio telefonico: la suoneria funziona regolarmente in caso di chiamata, mentre risulta in ogni caso escluso l'amplificatore telefonico.
- In C il microtelefono è sollevato per chiamare un utente o per essere stati chiamati. La sezione fonica del circuito è inserita (P1 chiuso e P2 aperto). Si può quindi ascoltare l'interlocutore che ha chiamato, o il segnale di centrale "libero" nell'altro caso.
- In D la configurazione di PD, PR e PI permette che anche in questa fase l'apparecchio telefonico funzioni secondo lo schema elettrico originario. In più

il suono è amplificato da AP, permettendo l'ascolto collettivo della chiamata o del segnale di centrale.
- In E si sta formando un numero sul disco combinatore. P2 esclude la sezione fonica dell'apparecchio. P1 si apre e si chiude ritmicamente in funzione del numero formato.
- In F è riproposta la stessa situazione con il dispositivo inserito. PR disconnette l'amplificatore telefonico, mentre PD invia in linea il codice della cifra formata.

Una volta formato l'intero numero telefonico, l'apparecchio ritorna nello stato illustrato in C e D, e perciò si ascolta il segnale di chiamata o eventualmente la voce dell'interlocutore desiderato, se risponde.
Le configurazioni di PD, PR e PI sono state riassunte per comodità in Figura 5.

Interfacciamento

Lo schema elettrico del dispositivo realizzato è visibile in Figura 6 e sostituisce la parte chiusa in tratteg-

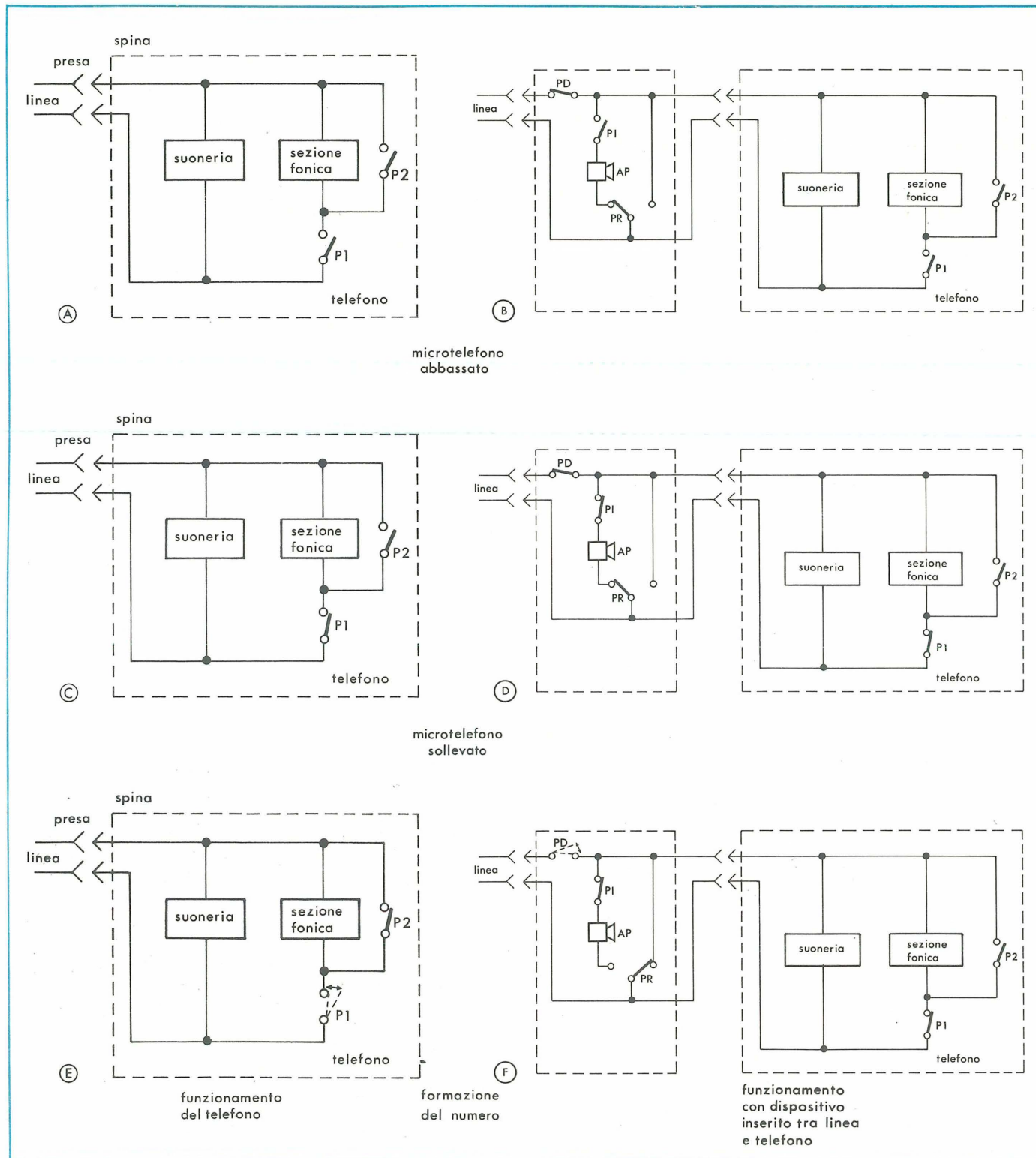



Figura 4 - Funzionamento del telefono.

Stato telefono	PD	PR	PI
Microtelefono abbassato	Chiuso	Aperto	Aperto
Microtelefono sollevato	Chiuso	Aperto	Chiuso
Formazione del numero	Aperto/Chiuso	Chiuso	Non importa

Figura 5 - Configurazioni di PD, PR, PI nei tre stati di funzionamento del telefono.

gio in B, D ed F di Figura 4. D1, D2, D3 e D4 esplicano una funzione esclusivamente protettiva. Infatti, anche ammettendo che vengano scambiati tra loro i due capi del doppino telefonico, essi provvederanno a fornire le giuste polarità al dispositivo, che di per sè è polarizzato facendo uso di transistors.

DZ1 fa in modo che eventuali sovratensioni presenti sulla linea, ad esempio durante l'azionamento della suoneria, non si presentino alla parte elettronica più sensibile (TR1, TR2, TR3).

La funzione del pulsante PD è affidata ai transistors TR1 e TR2. Collegati in configurazione Darlinghton, questi vanno in saturazione non appena si connette la R2 ad una tensione a livello logico 1TTL (circa 3 volt).

PI è realizzato con R1 e TR3. Non ha alcun collegamento di comando, in quanto il suo funzionamento è automatico. Infatti, non appena circola corrente in linea, TR3 va in saturazione, presentandosi una tensione ai capi di R1.

Riguardando la Figura 5, si può notare che PI deve essere aperto solo con microtelefono abbassato, quando cioè in linea non circola corrente. La disposizione circuitale di R1, TR3 fa in modo che questo si verifichi.

PR è realizzato con R3, TR4, RL1 e D5.

Una tensione a livello logico 1 TTL presentata ad R3 porta in saturazione TR4, il quale aziona il relé. D5 protegge TR4 dalle sovratensioni generate da RL1 negli istanti di commutazione.

Lo schema elettrico dell'amplificatore telefonico è in Figura 7.

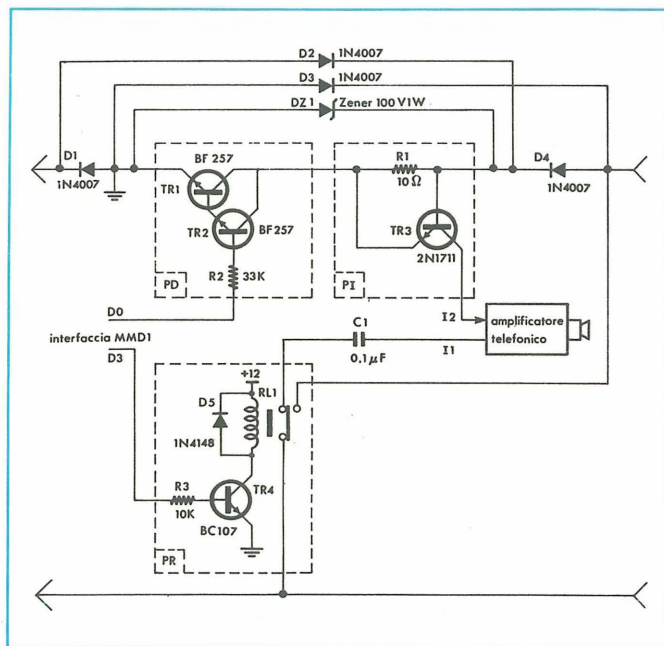



Figura 6 - Schema elettrico del combinatore telefonico.

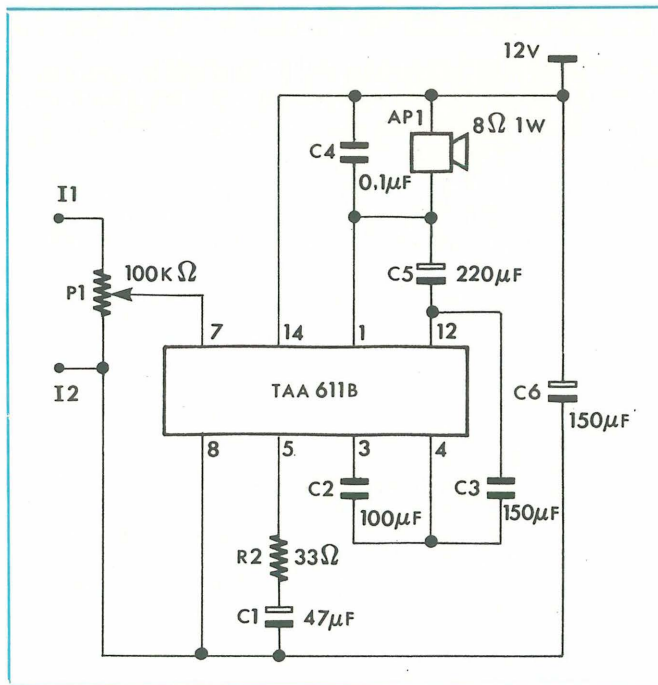



Figura 7 - Schema elettrico dell'amplificatore telefonico.

L'assorbimento è, a piena potenza, di circa 100 mA. Il segnale fonico, regolato dal potenziometro logaritmico P1, è portato tramite I1 al piedino d'ingresso dell'amplificatore. I2 invece è connesso al piedino di massa.

PI e PR provvedono a collegare correttamente questi due ingressi dell'amplificatore alla linea al momento opportuno.

Il comando di PD e PR è affidato ad un microelaboratore MMD1, il quale, in condizione di riposo del telefono, mantiene sempre azionato PD e diseccitato PR. Al momento di invio su linea di un numero telefonico, l'MMD1 eccita PR e aziona ritmicamente, a seconda del numero, PD.

Modalità operative

L'MMD1, oltre ad inviare un numero sulla linea attraverso il nostro dispositivo, dovrebbe avere associato un organo di Input/Output per permettere la realizzazione delle funzioni di memorizzazione di più numeri di abbonati ed una eventuale loro variazione, oltre alla visualizzazione degli stessi numeri nell'istante in cui essi sono formati. Questo dispositivo di ingresso/uscita potrebbe avere l'aspetto rappresentato in Figura 8.

I suoi elementi costitutivi sono:

1) Tasti elenco.
   Questi tasti indicano al microelaboratore in quale posizione vogliamo memorizzare il numero, oppure quale dei numeri, in precedenza memorizzati, vogliamo chiamare.
2) Tastiera decadica.
   Immette i numeri in memoria in fase di scrittura.
3) Tasto chiamata.
   Informa il microelaboratore che desideriamo chiamare uno dei dodici abbonati memorizzati.
4) Tasto scrittura in memoria.
   Informa il microelaboratore che desideriamo memorizzare un numero.

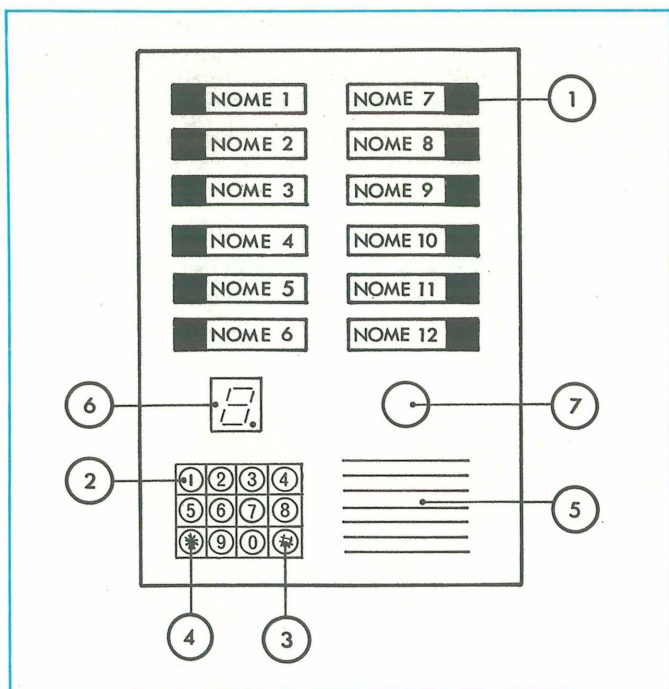



Figura 8 - Console del combinatore telefonico.

5) Altoparlante.
   E' connesso all'amplificatore ed entra in funzione appena si solleva il microtelefono, amplificando la voce dell'utente chiamato.
6) Display.
   Visualizza i numeri telefonici uscenti o entranti per consentire un controllo del numero composto.
7) Potenziometro di volume.
   Agisce sull'amplificatore, permettendo un maggiore o minore livello sonoro dell'altoparlante.

Le sequenze di operazioni sui tasti di comando sono le seguenti:
Per memorizzare un numero:

1) Premere il tasto contrassegnato dal simbolo *
2) Selezionare il tasto elenco desiderato
3) Impostare il numero tramite la tastiera decadica.

Per chiamare un numero di abbonato in precedenza memorizzato:

1) Premere il tasto di chiamata contrassegnato dal simbolo #
2) Premere il tasto elenco relativo all'abbonato che si intende chiamare.

Il software del combinatore telefonico

Lo schema a blocchi del software sviluppato è visibile in Figura 9.
Come si può notare, il flusso è, in linea di principio, molto semplice.
Dall'esame del tasto premuto il microelaboratore comprende l'operazione che deve eseguire:

- o procedere alla memorizzazione di un numero telefonico
- o al suo invio sulla linea.

Seguendo il listing del programma in Figura 12, si può vedere che, in questa fase iniziale:

1) Il bit D0 della porta 2 viene portato a livello logico 1 allo scopo di chiudere l'interruttore elettronico PD costituito dai due transistors TR1-TR2. Il telefono è ora connesso con la linea telefonica (stato B di Figura 4).
2) Tramite la subroutine di input KBRD, viene letta la tastiera decadica.
3) E' confrontato il contenuto dell'accumulatore (tasto appena letto) con il numero ottale 017 (codice del tasto #).
   Se (A) = 017, il microelaboratore si predispone all'invio in linea di un numero memorizzato in precedenza. Diversamente si procede in sequenza.
4) E' confrontato il contenuto dell'accumulatore con il numero ottale 016 (codice del tasto *).
   Se (A) = 016, il microelaboratore deve saltare alla sezione di programma adibita alla memorizzazione di un numero.
5) Se (A) ≠ 016, il programma ritorna al punto 2), in quanto è stato premuto un tasto non previsto.

"Come si può vedere, l'acquisizione di un dato da tastiera avviene tramite un sottoprogramma. Benché nella memoria ROM dell'MMD1 sia disponibile una subroutine che legge la tastiera, abbiamo preferito svilupparne una simile, che però acquisisce il codice del tasto al rilascio". Questo sottoprogramma, chiamato KBRD, ritorna in accumulatore il codice del tasto premuto.
L'intero programma del combinatore è stato posto nella memoria RAM, a partire dall'indirizzo 3000_8. I numeri telefonici sono invece memorizzati in RAM a partire dall'indirizzo 2006_8.

Memorizzazione di un numero telefonico

Supponiamo ora che sia premuto il tasto con il simbolo *, in quanto si vuole memorizzare un nuovo numero telefonico. Effettuata questa operazione, si deve poi specificare in quale dei dodici tasti elenco si intende memorizzare detto numero.
E' opportuna a questo punto una precisazione.
Tenendo presente che tutto il lavoro svolto ha valore puramente sperimentale, si è ritenuto non riduttivo usare la stessa tastiera decadica al posto della tastiera "elenco abbonati": in particolare la tastiera dell'MMD1 è utilizzata come unico organo di input. Nella realizzazione pratica quindi non ha assunto l'aspetto di Figura 8. La corrispondenza fra i tasti della tastiera dell'MMD1 ed il significato assunto nelle varie fasi di operatività è in Figura 10. L'aver associato al tasto G dell'MMD1 il valore numerico 0 è dovuto alla necessità di inviare in linea 10 impulsi senza ulteriori calcoli.
Continuiamo ora l'esame del programma.
Dopo aver premuto il tasto di memorizzazione nuovo numero, il programma interpreta il successivo input come numero progressivo dell'abbonato, per cui salta alla subroutine RCR. Questa converte ciascun codice di tasti elenco in una locazione di memoria. Segue quindi un'ulteriore fase di lettura della stessa tastiera, interpretata però come formazione del numero dell'abbonato prima definito, e come tale il numero è memorizzato nella locazione di memoria individuata in precedenza. Durante questa fase è fatto il confronto volta per volta tra il contenuto dell'accumulatore (che contiene la cifra impostata) ed il valore

013. Se dal confronto risulta che (A) ≥013, significa che è stato premuto o un tasto errato o un tasto di comando, in quanto al tasto * corrisponde il codice 016, ed al tasto # il codice 017, e quindi si ritorna allo stato iniziale. In caso contrario il programma procede in sequenza, provvedendo a visualizzare il numero introdotto sul display a 7 segmenti FND 500 tramite il sottoprogramma DISPLAY, che converte il codice della cifra introdotta nel codice a 7 segmenti. Infine è incrementato l'indirizzo della cella di memoria che accoglie il numero dell'abbonato, per predisporsi alla prossima cifra; quindi si ritorna alla fase di attesa di una nuova cifra.

Ad un più attento esame del listing si può vedere che in effetti si è voluto complicare leggermente il programma portandolo a memorizzare due cifre in una stessa cella di memoria. (Non chiedeteci se ne sia valsa la pena, ma ci si era preso tanto gusto a programmare, che non abbiamo potuto trattenerci dal farlo).

Possiamo evidenziare dall'esame del listing tre cose:

1) E' memorizzato anche il codice del tasto comando premuto dopo l'intero numero dell'abbonato. Esso verrà interpretato dal microelaboratore durante la fase di lettura del numero dell'abbonato

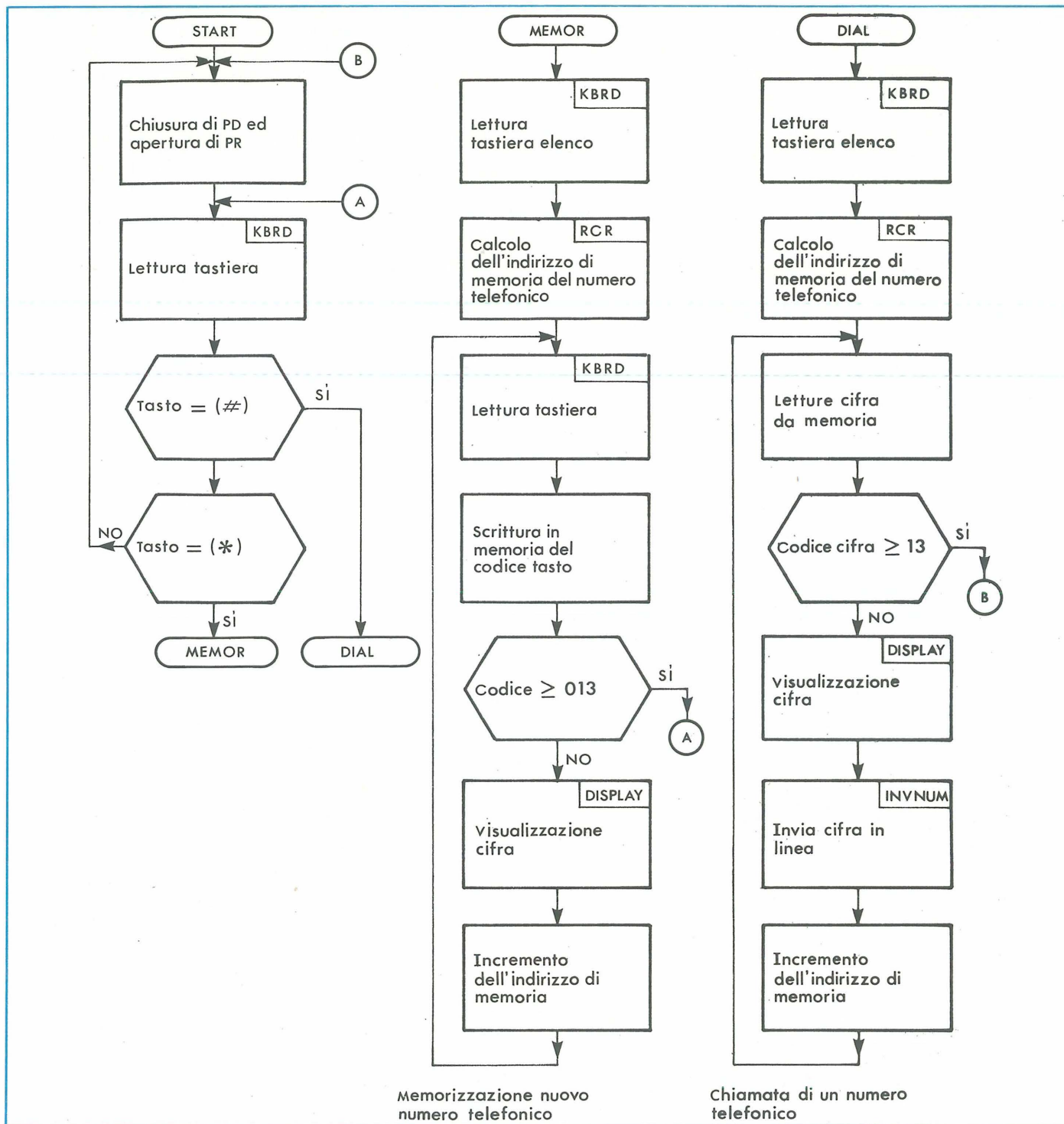



Figura 9 - Schema a blocchi del software del combinatore telefonico.

I "TANTI" PRODOTTI OK PER IL WIRE WRAPPING

wire wrapping center

SPELLA AVVOLGE SVOLGE

Avvolge, Spella e Svolge filo di AWG 30 (0,25 mm) su terminali quadrati da 0,63 mm di lato

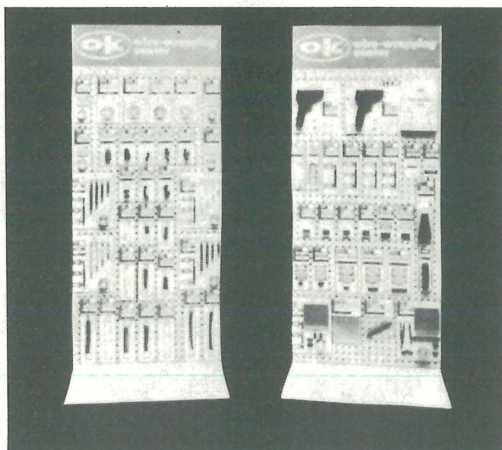
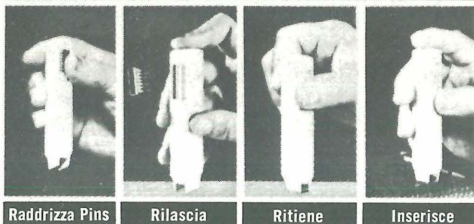
ATTREZZO "HOBBY WRAP"

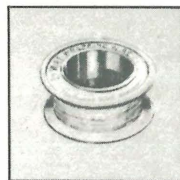
Wrappatura regolare	WSU-30
Wrappatura modificata	WSU-30 M

ATTREZZO A BATTERIA modello BW-630

completo di Punta e Copri Punta PER TERMINALI di 0,63 mm di lato

PER AWG 30	BW-630
PER AWG 26/28	BW-2628

PUNTE DI RICAMBIO	BT-30
PUNTE DI RICAMBIO	BT-2628

INSERITORE DI DIP/IC

Raddrizza Pins | Rilascia | Ritiene | Inserisce

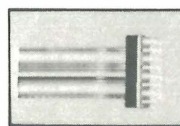
- Modello INS-1416

BOBINE DI FILO

Filo per Wire-Wrapping
AWG 30 KYNAR
15 mt di conduttore solido ricoperto d'argento, facile da spellare.
Disponibile nei colori: Blue, Bianco, Giallo e Rosso.

DISPENSATORE DI FILO

- Spella automaticamente il filo per 25,4 mm
- Taglia Filo Incorporato
- Ricaricabile

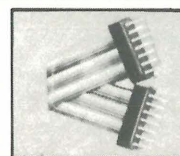
ESTRATTORE modello EX-1

Ideale per hobbisti e tecnici di laboratorio.
Può estrarre tutti i LSI-MSI-SSI da 8 a 24 Pin.

ZOCCOLI ASSEMBLATI da un lato solo con RIBBON CABLE

- Cavo piatto da 26 AWG (0,40 mm) lungo 609 mm

Modello SE 14-24	con 14 Pin Dip Plug
Modello SE 16-24	con 16 Pin Dip Plug

KIT per Wire Wrapping modello WK-4B

Contiene:
Attrezzo WSU-30 M
Dispensatore di filo WD-30-B
Due DIP da 14 Pin e due da 16 Pin
Una scheda hobby H-PCB-1
Un inseritore INS-1416
Un estrattore EX-1
Un connettore PCB modello CON-1

ZOCCOLI ASSEMBLATI con RIBBON CABLE da entrambe i lati

Con 14 Pin Dip Plug lungo 50,8 mm	DE 14-2
Con 14 Pin Dip Plug lungo 203,2 mm	DE 14-8
Con 16 Pin Dip Plug lungo 101,6 mm	DE 16-4

* Numerosi altri modelli a Stock

OK MACHINE & TOOL ITALIA S.R.L.

VIA CARLO RAVIZZA 34/1 - 20149 MILANO - TEL. 02/4983573 - TELEX 320419 BRUS I

Tasto MMD1	Codice ottale	Numero memorizzato	Numero progressivo abbonato	Tasti di comando
1	001	n. 1	1° abbonato	—
2	002	n. 2	2° abbonato	—
3	003	n. 3	3° abbonato	—
4	004	n. 4	4° abbonato	—
5	005	n. 5	5° abbonato	—
6	006	n. 6	6° abbonato	—
7	007	n. 7	7° abbonato	—
H	010	n. 8	8° abbonato	—
L	011	n. 9	9° abbonato	—
G	012	n. 0	10° abbonato	—
S	013	—	11° abbonato	—
A	015	—	12° abbonato	—
B	016	—	—	* memorizzazione nuovo
C	017	—	—	# numero chiamata di un numero

Figura 10 - Significato dei tasti dell'MMD1 durante le varie fasi di operatività.

come segno di "fine numero". In ogni caso, comunque, terminato di impostare l'ultima cifra del numero dell'abbonato da memorizzare, si ritorna allo stato iniziale (esecuzione nuovo comando), che è aperto dalla pressione di * o di #.

2) Non esiste alcun controllo sul numero massimo delle cifre da memorizzare. Si è previsto comunque che non siano più di 11 (4 di prefisso e 7 di numero dell'abbonato). Attenti quindi a non pigiare senza fine sulla tastiera!

3) Non si controlla se l'abbonato ha già un numero o no. L'operazione di prima composizione del nostro elenco elettronico abbonati è quindi identica all'operazione di *variazione* del numero stesso. Quindi, se ci accorgiamo di aver sbagliato ad introdurre una cifra, fermiamoci, riprendiamo a battere il tasto * (B nell'MMD1), e ricominciamo a formare il numero corretto.

In Figura 11 è riportata una possibile configurazione della zona di memoria destinata a ritenere l'elenco degli abbonati.

Figura 11 - Elenco telefonico in memoria.

Chiamata di un numero telefonico

Partendo dallo stato iniziale e premendo il tasto contrassegnato dal simbolo #, si informa il microelaboratore che si intende chiamare un numero già memorizzato. Si procede poi, secondo la sequenza già indicata, alla pressione del tasto elenco (nel nostro "povero" dispositivo la stessa tastiera decadica) corrispondente all'utente col quale si intende comunicare.

In seguito a questa sequenza, che il microelaboratore interpreta come richiesta di chiamata di un numero, avviene quanto segue:

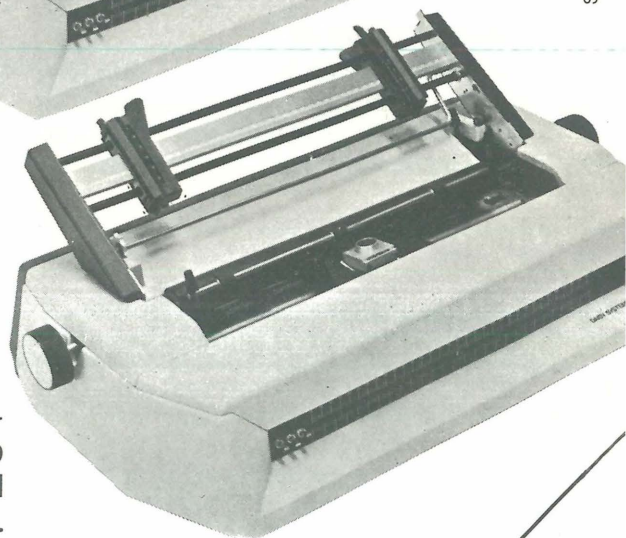
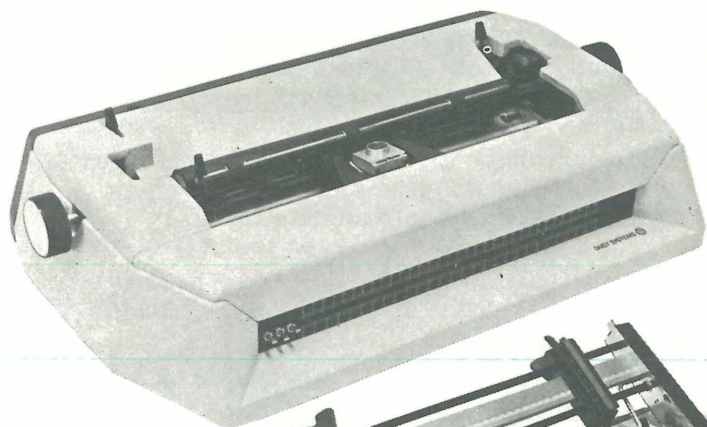
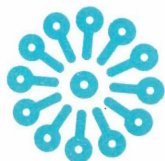
1) Il programma salta alla subroutine RCR. Come già detto, essa converte ciascun codice dei tasti elenco in un indirizzo di memoria.
2) Il contenuto della locazione di memoria (indicata dai registri H ed L) va nell'accumulatore.
3) Il byte presente ora in A contiene due cifre del numero da chiamare. Viene considerata solo quella il cui codice si trova nei primi quattro bit (D0 $\div$ D3).
4) Confronto tra (A) ed il byte 013.
   Se (A) $\geqslant$ 013, significa che il numero è stato formato integralmente, per cui si torna allo stato iniziale. Viceversa, se (A) $<$ 013, si prosegue in sequenza.
5) Si visualizza la cifra chiamata tramite la subroutine DISPLAY.
6) Ponendo in uscita sulla porta 2 il byte 010, si blocca la parte fonica (PR in Figura 6) e si apre l'interruttore elettronico PD. Quest'ultimo resta aperto per 40 msec, chiudendosi poi per una durata di 60 msec. L'operazione si ripete per un numero di volte pari alla cifra da trasmettere (se 6, sei volte). Al contrario la parte fonica resta bloccata per tutta la durata della composizione di ciascuna cifra.
   Una volta inviata una cifra, è previsto un ritardo di 800 msec prima che si possa trasmettere la successiva.
7) Ora può essere trasmessa, se esiste, la seconda cifra contenuta nella locazione di memoria indicata da H ed L, cioè quella il cui codice si trova nei bit D4 $\div$ D7. Per far questo, riletta la cella di memoria in A, il contenuto di A viene fatto ruotare di quattro posizioni verso destra. Dopo aver incrementato il registro HL, si torna al punto 3).

N.B. I tempi di apertura e chiusura sono regolati da un'apposita subroutine, TIMEOUT, che è pure presente nella KEX. La routine TIMEOUT genera un ritardo di 10 msec.

Aggiungi... un pezzo al tavolo

La Daisy è una stampante a margherita versatilissima: è sufficiente infatti cambiare la margherita per avere disponibili svariati tipi di caratteri. L'interfaccia parallela (TTL) o quella Qume compatibile, e la RS 232 disponibile come opzione, la rendono interfacciabile con una vasta gamma di macchine. La stampa piena e perfetta fa della Daisy la stampante ideale per tutti i livelli cui è richiesta una scrittura particolarmente curata, nel settore del Word-Processing.

Progettata per un'ampia varietà di applicazioni, la Daisy Wheel Printer M-50 può lavorare anche come plotter per la capacità di spostamento carta nei due sensi con caratteristica di battuta per punti ad alta velocità. Con i suoi trascinamoduli a inseritore frontale di scheda, la M-50 offre ai piccoli sistemi multiusi, possibilità che si possono trovare oggi soltanto in sistemi a costi più elevati. Inoltre, grazie ai suoi servomeccanismi basati su motori standard, può essere usata come un dispositivo di stampa ad alta velocità nella nuova generazione di macchine intelligenti per scrivere.

Studio Cambiaghi - Milano

Basso rumore, nastro a cartuccia, impatto del martelletto sincronizzato consentono elevate prestazioni ad un prezzo particolarmente contenuto.

...e in più vi diamo una mano grossa così.

DAISY SYSTEMS
è rappresentata in Italia da:

20124 MILANO - Via Timavo, 12
Tel. (02) 6073184 - 6073255 - 607088 - 692882

Desidero ricevere informazioni su Daisy

Ditta ....

Nome ....

Via ....

Tel. ....

Cap ....

Città ....

Sottoprogrammi utilizzati

Subroutine KBRD

La subroutine KBRD ha alcune caratteristiche che illustrano un'interessante correlazione hardware-software. I pulsanti usati nel microelaboratore MMD1 non sono privi di rimbalzi, per cui, quando gli switches sono commutati, esiste un breve periodo di tempo durante il quale i contatti non risultano essere collegati tra loro in modo stabile. Questo può creare confusione al computer, dato che esso non è in grado di distinguere fra una chiusura reale del tasto ed un rimbalzo. La subroutine KBRD evita questo inconveniente in quanto riconosce qualunque pressione dei tasti ma non ricava il codice del tasto se non dopo aver verificato che il tasto è stato premuto effettivamente.

Quanto detto è realizzato attendendo un breve periodo di tempo (10 msec) dopo aver avvertito la pressione sul tasto, e ricontrollando poi che il tasto sia ancora premuto.

Analogo controllo è fatto sul rilascio del tasto, ossia si verifica che il tasto ha smesso di rimbalzare prima di analizzare se è premuto un altro tasto.

Subroutine TIMEOUT

La subroutine TIMEOUT non influenza nessuno dei registri o dei flags, e serve solo per creare un ritardo di tempo. Con un microprocessore 8080A che opera a 750 kHz, questa subroutine genera un ritardo di 10 msec.

Subroutine RCR

Questa subroutine, a differenza della precedente, che si trova nella memoria di sola lettura inserita in KEX, è stata da noi elaborata per la nostra applicazione. Essa converte ciascun codice dei tasti elenco in una locazione di memoria. Poiché ad ogni numero telefonico ne sono riservate sei, premendo il tasto elenco 1, la subroutine caricherà nel registro L il byte 006; premendo il tasto elenco 2, caricherà il byte 014, e così via.

Riassumendo, nella subroutine RCR entra il codice del tasto elenco ed esce l'indirizzo della prima posizione di memoria relativa al numero da memorizzare o memorizzato in precedenza.

Subroutine DISPLAY

Questa subroutine dimostra come sia possibile sostituire a volte l'hardware col software. Infatti essa sostituisce dal punto di vista funzionale una decodifica 7447 o 7448. La subroutine trasforma il codice BCD di una cifra nel codice a 7 segmenti utilizzato dal display.

■

```
;*************************************************************
;
;
;           COMBINATORE TELEFONICO
;
;************************************************************
;
;
;
;
;*************  MONITOR TASTIERA  *********************
;
;
START  MVI    A,01     ; CHIUSURA DI PD ED APERTURA DI PR
       OUT    02
       CALL   KBRD     ; INPUT CODICE TASTO
CONF   CPI    017
       JZ     DIAL     ; SE CODICE = 017 ESEGUI DIAL
       CPI    016
       JZ     MEMOR    ; SE CODICE = 016 ESEGUI MEMOR
       JMP    START    ; SE ALTRO CODICE RITORNA A START
;
;
;
;***********  MEMORIZZAZIONE DI UN NUMERO TELEFONICO  ************
;
;
MEMOR  CALL   KBRD     ; INPUT CODICE ELENCO
       CPI    016
       JNC    CONF     ; SE CODICE > = 016 RITORNA A MONITOR (CONF)
       CALL   RCR      ; TRASFORMA CODICE IN INDIRIZZO
INPUT  CALL   KBRD     ; INPUT CODICE CIFRA NUMERO TELEFONICO
       MOV    M,A      ; MEMORIZZA CIFRA
       CPI    013
       JNC    CONF     ; SE CODICE CIFRA > = 013 RITORNA A MONITOR (CONF)
       CALL   DISPLAY  ; VISUALIZZA CIFRA
       CALL   KBRD     ; INPUT CODICE CIFRA NUMERO TELEFONICO
       MOV    B,A
       RLC
       RLC
       RLC
       RLC
       ORA    M
       MOV    M,A      ; MEMORIZZA CIFRA AL MEDESIMO INDIRIZZO
       INX    H        ; INCREMENTA INDIRIZZO MEMORIA
       MOV    A,B
       CPI    013
       JNC    CONF     ; SE CODICE CIFRA > = 013 RITORNA A MONITOR (CONF)
       CALL   DISPLAY  ; VISUALIZZA CIFRA
       JMP    INPUT    ; RITORNA A INIZIO CICLO MEMORIZZAZIONE CIFRA
;
;
;
;*************  CHIAMATA DI UN NUMERO TELEFONICO  ***************
;
;
DIAL   CALL   KBRD     ; INPUT CODICE ELENCO
       CPI    016
       JNC    CONF     ; SE CODICE > = 016 RITORNA A MONITOR (CONF)
       CALL   RCR      ; TRASFORMA CODICE IN INDIRIZZO
DIAL2  MOV    A,M      ; LEGGI DA MEMORIA COPPIA CIFRE
       MVI    B,002
       ANI    017
       MOV    E,A      ; RICAVA IN (E) PRIMA CIFRA
DIAL3  CPI    013
       JNC    START    ; SE CODICE > = 013 RITORNA A MONITOR (START)
       CALL   DISPLAY  ; VISUALIZZA CIFRA
       CALL   INVNUM   ; INVIA CIFRA IN LINEA
       DCR    B
       JZ     DIAL2    ; DOPO INVIO COPPIA DI CIFRE RITORNA A CICLO LETTURA MEMORIA
       MOV    A,M      ; RILEGGI DA MEMORIA COPPIA CIFRE
       RRC
       RRC
       RRC
       RRC
       ANI    017
       MOV    E,A      ; RICAVA IN (E) SECONDA CIFRA
       INX    H        ; INCREMENTA INDIRIZZO MEMORIA
       JMP    DIAL3    ; RITORNA A CICLO INVIO CIFRA IN LINEA
;
;
;
;
;****************  SUBROUTINES UTILIZZATE  ***************
;
;
;****************************************
;                                       *
;  INVNUM : INVIO CIFRA IN LINEA.       *
;           LA CIFRA E" IN (E).         *
;                                       *
;****************************************
;
INVNUM MVI    A,10
       OUT    002      ; CHIUSURA DI PR ED APERTURA DI PD PER 40 MSEC
       MVI    C,004
MS40   CALL   TIMEOUT  ; ESECUZIONE DI UNA ATTESA DI 10 MSEC
       DCR    C
       JNZ    MS40     ; RIPETI ATTESA 4 VOLTE
       MVI    A,011
       OUT    002      ; CHIUSURA DI PD PER 60 MSEC
       MVI    C,006
MS60   CALL   TIMEOUT  ; ESECUZIONE DI UNA ATTESA DI 60 MSEC
       DCR    C
       JNZ    MS60     ; RIPETI ATTESA 6 VOLTE
       DCR    E        ; DECREMENTA CIFRA DA INVIARE
       JNZ    INVNUM   ; SE SEQUENZA INVIO NON ESAURITA RIPETI CICLO
       MVI    A,001
       OUT    002      ; APERTURA DI PR E CHIUSURA DI PD PER 800 MSEC
       MVI    C,100
MS640  CALL   TIMEOUT  ; ESECUZIONE DI UNA ATTESA DI 10 MSEC
       DCR    C
       JNZ    MS640    ; RIPETI ATTESA 64 VOLTE
       MVI    A,000
       OUT    000      ; SPEGNI DISPLAY
       MVI    C,012
MS100  CALL   TIMEOUT  ; ESECUZIONE DI UNA ATTESA DI 10 MSEC
       DCR    C
       JNZ    MS100    ; RIPETI ATTESA 10 VOLTE
       RET
;
;
;
;
;******************************
;                            *
;     RCR : TRASFORMAZIONE DI UN   *
;           CODICE IN INDIRIZZO.   *
;           IL CODICE E" IN (A)    *
;                            *
```

```
;                                    *
;*************************************
;
;
RCR     LHI     H,2000  ;(HL) = INIZIO MEMORIA ELENCO - 6
        LXI     B,006
RCR1    DCR     A
        RM
        DAD     B       ; SOMMA AD (HL) 6
        JMP     RCR1
;
;
;
;****************************************
;                                       *
;    DISPLAY : VISUALIZZAZIONE CIFRA    *
;              LA CIFRA E" IN (A).      *
;                                       *
;****************************************
;
;
DISPLAY PUSH    H
        LHI     H,SEGM7
DIS1    DCR     A
        JZ      DIS2
        INX     H
        JMP     DIS1    ; CALCOLO INDIRIZZO CODICE 7 SEGMENTI PER LA CIFRA IN (A)
DIS2    MOV     A,M     ; LEGGI CODICE 7 SEGMENTI INDIVIDUATO
        OUT     000     ; OUTPUT CODICE
        POP     H
        RET
;
;
;******  TABELLA CONVERSIONE CIFRA CODICE 7 SEGMENTI  ******
;
SEGM7   BYTE    104     ; CODICE 7 SEGMENTI PER IL NUMERO 1
        BYTE    153     ; CODICE 7 SEGMENTI PER IL NUMERO 2
        BYTE    156     ; CODICE 7 SEGMENTI PER IL NUMERO 3
        BYTE    134     ; CODICE 7 SEGMENTI PER IL NUMERO 4
        BYTE    076     ; CODICE 7 SEGMENTI PER IL NUMERO 5
        BYTE    037     ; CODICE 7 SEGMENTI PER IL NUMERO 6
        BYTE    144     ; CODICE 7 SEGMENTI PER IL NUMERO 7
        BYTE    177     ; CODICE 7 SEGMENTI PER IL NUMERO 8
        BYTE    174     ; CODICE 7 SEGMENTI PER IL NUMERO 9
        BYTE    167     ; CODICE 7 SEGMENTI PER IL NUMERO 0
;
;
;
;*****************************************
;                                        *
;   TIMEOUT : REALIZZA UNA ATTESA DI 10  *
;             MSEC CON CLOCK DI 750 KHZ  *
;                                        *
;*****************************************
;
;
TIMEOUT PUSH    PSW
        PUSH    D
        LXI     D,146
TIMOU1  DCX     D
        MOV     A,D
        ORA     E
        JNZ     TIMOU1
        POP     D
        POP     PSW
        RET
;
;
;
;*****************************************
;                                        *
;   KBRD : LETTURA CARATTERE DA TASTIERA *
;          IL CODICE SI OTTIENE IN (A)   *
;                                        *
;*****************************************
;
;
KBRD    IN      000     ; INPUT DAL KEYBOARD ENCODER
        ORA     A
        JM      KBRD    ; SE (A) < 0 CONTINUA A LEGGERE IL KEYB. ENCODER
        CALL    TIMEOUT ; TASTO PREMUTO. ATTENDI FINE RIMBALZI
KBR3    IN      000     ; DOPO 10 MSEC RILEGGI IL KEYB. ENCODER
        ORA     A
        JM      KBRD    ; SE (A) <0 TORNA ALL"INZIO PERCHE" DISTURBO
        MOV     C,A     ; SE (A) > 0 = 0 SALVA CODICE IN (C)
KBR1    IN      000     ; RILEGGI KEYB. ENCODER IN ATTESA RILASCIO TASTO
        ORA     A
        JP      KBR1
        CALL    TIMEOUT ; TASTO RILASCIATO. ATTENDI FINE RIMBALZI
        IN      000     ; DOPO 10 MSEC RILEGGI KEYB. ENCODER
        ORA     A
        JP      KBR3    ; SE (A) > 0 = 0 RIPETI CICLO RILASCIO TASTO
        MOV     A,C     ; RIMETTI IN (A) CODICE TASTO
        ANI     017
        PUSH    H
        MOV     C,A
        MVI     B,0
        LXI     H,TASCOD
        DAD     B       ; CALCOLO INDIRIZZO CODICE MMD1 CORRISPONDENTE A TASTO
        MOV     A,M     ; LEGGI DA TABELLA CONVERSIONE CODICE MMD1
        POP     H
        RET
;
;
;******  TABELLA CONVERSIONE CODICE TASTIERA IN CODICE MMD1  ******
;
;
TASCOD  BYTE    000     ; TASTO 0
        BYTE    001     ; TASTO 1
        BYTE    002     ; TASTO 2
        BYTE    003     ; TASTO 3
        BYTE    004     ; TASTO 4
        BYTE    005     ; TASTO 5
        BYTE    006     ; TASTO 6
        BYTE    007     ; TASTO 7
        BYTE    013     ; TASTO S
        BYTE    017     ; TASTO C
        BYTE    012     ; TASTO G
        BYTE    010     ; TASTO H
        BYTE    011     ; TASTO L
        BYTE    015     ; TASTO A
        BYTE    016     ; TASTO B
```

Figura 12 - Listing del software del combinatore telefonico.

Controllo di ferrovie in miniatura: un esempio

di **L. Furiozzi** - *General Processor, Firenze*

I sistemi personal computer a microprocessore, per i costi altamente contenuti e per l'estrema versatilità di impiego, stanno rapidamente portando al superamento di un atavico complesso d'inferiorità nei confronti del "calcolatore".
Guardati fino a poco tempo fa con sospetto, o con una sorta di deferenza, come qualcosa di estraneo e di lontano dalla pratica quotidiana, si stanno ormai imponendo in tutta una serie di campi come validi e preziosi strumenti di lavoro; le loro applicazioni scientifiche, professionali, commerciali e didattiche sono in continua crescita.
Inoltre, con il personal computer si può giocare: hanno iniziato ad utilizzarlo in tal senso i vecchi e fedeli radioamatori che, in Italia, lo hanno tenuto a battesimo e ne hanno guidato i primi, incerti passi; se ne sono ormai appropriati i molti consumatori di giochi televisivi, che vi hanno trovato una fonte inesauribile di giochi che essi stessi possono ideare.
Ma con un personal computer si può fare molto di più: ad esempio, l'hobby del modellismo ferroviario con l'abbinamento del personal computer acquista una dinamicità che diversamente non ha.
Infatti, una volta comprati i binari, i vagoni, le motrici, costruite altissime montagne e scroscianti cascate, inseriti semafori e passaggi a livello, montato ed as-

semblato tutto il plastico, di solito non rimane che far partire il treno lungo il percorso stabilito per poi starlo a guardare.
Oggi la presenza del microprocessore dà la possibilità di vestire i panni del capostazione, ovvero dirigere la ferrovia in miniatura; è sufficiente infatti compilare semplici programmi di calcolo che, se non sono stati commessi errori di programmazione, gestiscono il modellino ferroviario in maniera corretta.
Il "capostazione" può così affidarsi alla propria fantasia ed inventare, ogni volta che lo desidera, nuovi percorsi, nuove situazioni, senza dover operare modifiche circuitali sul plastico.
Qualcuno potrà chiedere: come fa il microprocessore a sapere quali sono i binari sui quali i treni stanno viaggiando, la loro velocità e direzione, come controllare che le sbarre del passaggio a livello siano alzate od abbassate, come inviare i comandi a tutte le unità interessate?
Sono domande più che lecite a cui si può rispondere facilmente: è sufficiente avere delle interfacce di trasduzione tali da far comprendere i segnali di controllo e di comando, provenienti dal microprocessore, al dispositivo periferico e viceversa (v. Fig. 1).
A questo punto accenniamo brevemente alle prestazioni più classiche di un modellino ferroviario ed ai dispositivi impiegati per realizzarle.
Supponiamo per semplicità di avere un solo treno, costituito da una motrice ed alcuni vagoni; quello che possiamo fare facilmente, senza grandi complicazioni circuitali, è:

- partenza in una determinata direzione,
- controllo della velocità del treno nei vari tratti di percorso,
- sgancio di vagoni,
- accensione di semafori,
- controllo della posizione delle sbarre del passaggio a livello,
- sincronizzazione di eventi in base a liste preordinate.

Vediamo ora un esempio pratico.
Supposto di aver montato il plastico, completati i collegamenti necessari (v. Fig. 2), non rimane altro che

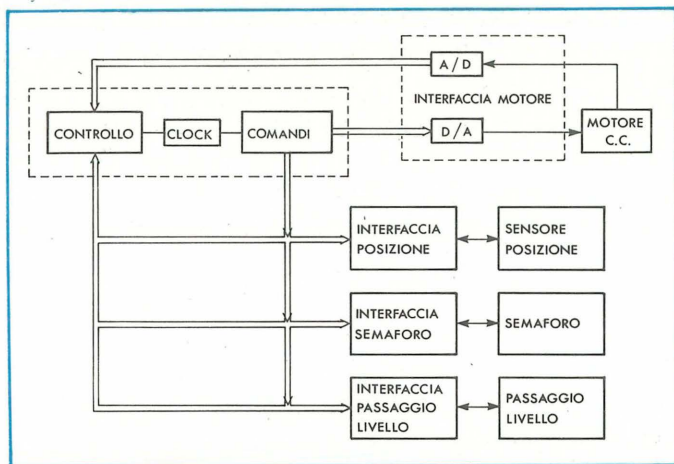



Figura 1 - Schema dell'interazione di un microprocessore con un modellino ferroviario.

scrivere il programma che consenta al treno di viaggiare come desiderato.
Lungo i binari sono posti dei sensori di posizione che, scattando al passaggio del treno, ne rilevano la posizione; quest'ultima, tramite un'opportuna interfaccia, è comunicata al sistema di controllo del personal computer (v. Fig. 3).
Definita la tabella di viaggio (v. Tabella I), il compito del capostazione consiste nello scrivere il programma ed inserire i dati nella memoria del computer.
Il programma può essere scritto in un linguaggio come il BASIC in cui i comandi di istruzione e di programmazione sono di facile comprensione; sono sufficienti infatti poche ore per imparare a programmare in BASIC.
In Fig. 4 è riportato lo schema a blocchi della procedura di calcolo che realizza la tabella di viaggio prescelta. Il percorso, ovvero il numero dei passaggi a livello e dei semafori che il treno incontra, i sensori di posizione interessati (per semplicità sono numerati sequenzialmente) e la corrispondente potenza da erogare al motore possono essere accettati, come dati di programma, con l'istruzione INPUT, battendo i valori scelti sulla tastiera della console, terminando con il ritorno carrello.
Es.:

```
50 PRINT "NUMERO SEMAFORI INCONTRATI" ....
60 INPUT NS
.
.
```

Figura 2 - Percorso del treno. 1, 2 .. N numero di sensori di posizione presenti lungo il tragitto.

L'istruzione PRINT fa stampare sullo schermo video della console la frase scritta fra virgolette, di evidente aiuto mnemonico per l'operatore.
A questo punto il microprocessore si 'interroga' circa lo stato del treno e, in base alla risposta ricevuta, il programma salta ad istruzioni diverse.
Il treno può essere fermo o perché non é ancora partito (NSE = 0) o perché è arrivato in stazione (NSE = 15), o perché è uscito fuori dai binari per cause impreviste (in quest'ultimo caso, sullo schermo video, appare un segnale d'allarme) od infine perché in attesa di un semaforo rosso.
L'istruzione REM inserisce commenti nel listing del programma, che sono d'aiuto all'operatore, ma che il microprocessore non vede.
Es:

```
.
.
20 REM TRENO FERMO IST = 0
30 REM TRENO IN MOVIMENTO IST = 1
40 REM NSE = NUMERO SENSORE
       INTERESSATO
50 IF IST= 0 THEN 110
.
.
90  REM INIZIO PERCORSO NSE = 0
100 REM SEMAFORO NSE = 8
110 IF NSE = 0 THEN 300
120 IF NSE = 15 THEN 160
130 IF NSE = 8 THEN 250
140 PRINT "ALLARME LUNGO LA LINEA"
150 REM IFP = FINE PERCORSO
160 PRINT "FINE PERCORSO? SI = 0, NO ≠ 0"
170 INPUT IFP
180 IF IFP = 0 THEN 350
.
.
250 REM CONTROLLO SE IL SEMAFORO E'
        ROSSO
.
300 REM COMANDO DI PARTENZA
.
350 REM COMANDO DI STOP
.
```

Ogni volta che il treno passa in corrispondenza di un sensore di posizione, arriva un segnale al microprocessore che provvede al controllo del sensore interessato, cioè identifica il tratto di binario che il treno sta iniziando a percorrere e controlla la corrispondente velocità del motore, variandone la potenza

Figura 3 - Esempio di interfaccia. L'informazione relativa al passaggio del treno sul binario è tradotta in 0 (binario libero) e 1 (binario occupato). Il dispositivo periferico impiegato è un sensore di posizione costituito da un relé magnetico che scatta ogni qualvolta transita un vagone su cui è stato precedentemente fissato un magnete.

Tabella I - Tabella di viaggio.

Sensori di posizione	Potenza da erogare al motore del treno (valori puramente indicativi)
1 tratto piano	3
2 inizio salita	4
3 salita	5
4 salita	5
5 culmine salita	6
6 discesa	3
7 fine discesa	3
8 prima del passaggio a livello	2
9 dopo il passaggio a livello	2
10 in curva	1
11 fine curva	3
12 prima del semaforo, in piano	3 o 1
13 dopo il semaforo, in piano	3
14 in prossimità della stazione	2
15 in stazione	0 o 1

quando richiesto (accelerare in salita, decelerare in curva, nelle vicinanze della stazione, del semaforo se rosso, etc., etc.).
Inoltre, se il sensore è in corrispondenza di un passaggio a livello o di un semaforo, il microprocessore esamina lo stato di quest'ultimi e lo varia, se necessario o previsto dalla tabella di viaggio.

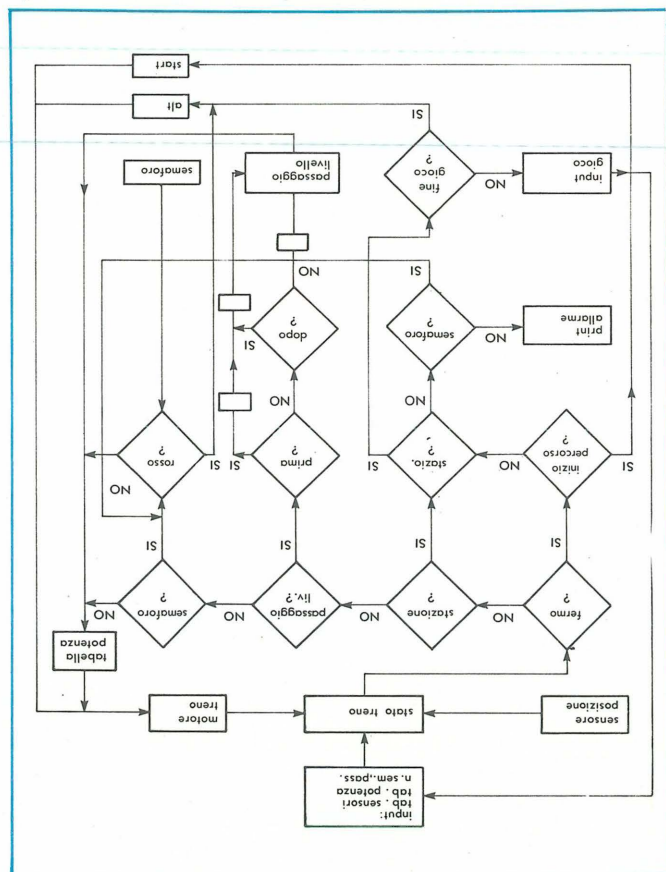



Figura 4 - Diagramma di flusso della procedura di calcolo che realizza la tabella di viaggio.

Concludendo, quanto sopra è stato scritto unicamente per dare un'idea delle possibilità di impiego del microprocessore applicato all'hobby del modellismo ferroviario.
Ai curiosi ed agli appassionati non rimane che provare; alla vostra fantasia si aprono nuove frontiere e vasti orizzonti da esplorare. Buon divertimento! ■

5144

Mastermind

Tratto da: "Basic Computer Games" (Creative Computer Press)*

Il Mastermind è uno dei più interessanti giochi di logica. Mastermind è marchio registrato della Invicta Plastics ed è in vendita sotto forma di una tavoletta (v. Figura 1) corredata da una serie di pezzi colorati raggruppati in sei colori, più due serie di pezzi bianchi e neri.
Programmare un personal computer per giocare a Mastermind è quindi, per chi lo conosce, un'esperienza da fare. Per chi non lo conosce, ecco i primi rudimenti.
Il mastermind è un gioco logico, nel quale intervengono la logica deduttiva, la verifica di ipotesi e la deduzione basata sulle probabilità. In America molte società di calcolatori lo usano come test per verificare le attitudini alla programmazione dei candidati all'assunzione.
E veniamo al gioco. I giocatori sono due: il primo è chiamato *code-maker*, il secondo *code-breaker*. Durante il gioco, ad ogni fine partita si possono scambiare i ruoli. Il *coder-maker* è così chiamato perché dispone di una serie di pezzi (quattro) secondo una certa combinazione che l'avversario deve indovinare (di qui il nome *code-breaker*). È chiaro che la combinazione di pezzi disposta dal code-maker è fatta in modo che l'avversario non la veda. Questi comunque cerca di indovinarla disponendo i propri mezzi secondo una data sequenza; in risposta riceve quindi dal codemaker il risultato delle proprie mosse: per ogni pezzo individuato e come colore e come posizione, un pezzo bianco.
Si noti pertanto che la risposta fornita dal codemarker non indica *quali* elementi sono stati indovinati, ma solo *quanti*.
Ad esempio, se il code-maker dispone i propri pezzi secondo la sequenza:

Giallo, Rosso, Rosso, Verde

ed il code-breaker ipotizza la seguente:

Rosso, Rosso, Giallo, Blu

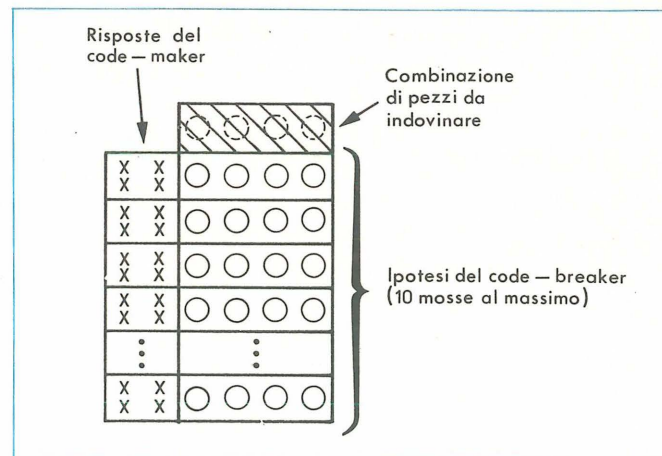
il code-maker gli darà in risposta due pezzi bianchi ed uno nero, per cui il code-breaker ignorerà qual'è il pezzo indovinato come colore e come posizione (a meno di non considerare i risultati ottenuti nelle mosse precedenti).
I colori dei pezzi di gioco sono solitamente: Rosso, Blu, Marrone, Verde, Giallo e Arancio. E' permesso di utilizzare lo stesso colore più di una volta nella sequenza. Il gioco prevede che ogni partita termini in 10 mosse, cioé il code-breaker ha a disposizione al massimo 10 combinazioni per indovinare la sequenza del code-maker e quindi vincere la partita.
Scrivere un programma BASIC per giocare a Mastermind vuol dire attribuire al computer o il ruolo di code-maker, nel qual caso noi dobbiamo indovinare la serie e la posizione dei quattro colori, o quello di code-breaker, nel qual caso il computer deve indovinare la nostra sequenza in base alle indicazioni di bianchi e neri che noi gli forniamo.
Il programma qui presentato prevede che ogni colore sia indicato con la lettera iniziale; avremo quindi:

Rosso = R
Blu = B
Marrone = M
Verde = V
Giallo = G
Arancio = A

Inoltre sono state apportate alcune varianti, nel senso che all'inizio del gioco è richiesto che venga precisato il numero di colori utilizzati (statement 80) e il numero di pezzi per ogni combinazione (statement 100). Comunque con questo programma il computer, oltre a svolgere il ruolo di giocatore, svolge un servizio di aiuto (*neutro*) per noi, in quanto:

1) fornisce il numero di combinazioni corrispondente alla coppia numero di colori-numero di pezzi impostata inizialmente;
2) quando noi giochiamo come code-breaker, possiamo chiedere la lista delle precedenti mosse scrivendo "TAVOLA", in qualsiasi momento (statement 390);
3) il computer tiene il conteggio delle mosse impiegate per indovinare la combinazione.

Consideriamo ora quella parte di programma che implementa il gioco vero e proprio. Distinguiamo due momenti: nel primo il computer ricopre il ruolo di code-maker (statements 370-610), nel secondo è il

* Distribuito in Italia dalla Jackson Editrice s.r.l.

computer che cerca di indovinare la nostra combinazione (statements 740-1080). L'implementazione della strategia di code-maker è molto semplice, in quanto si riduce all'istruzione RND (generazione di un numero casuale) contenuta nello statement 320. Più interessante è l'implementazione della strategia di code-breaker, anche se di fatto non presenta alcuna difficoltà.
Immaginiamo infatti di scrivere inizialmente la lista di *tutte* le combinazioni che si possono ottenere con un dato un numero di colori (⩽6). E con un dato numero di pezzi (⩽6).Se, per ogni risposta fornita dal calcolatore, cancelliamo dalla lista le combinazioni che risultano essere non più valide, saremo in grado di arrivare rapidamente alla soluzione. Se ad esempio, avendo proposto come ipotesi la sequenza Rosso, Rosso, Verde, Verde, non guadagnamo nessun pezzo, ne deduciamo che qualsiasi combinazione che contenga un pezzo rosso o un pezzo verde è impossibile, per cui la cancelleremo dalla lista. Si prosegue così finché si arriva alla soluzione esatta, o finché sulla lista non restano più combinazioni, nel qual caso evidentemente l'avversario ha commesso un errore nel fornire le risposte.
Ora implementare questa strategia, così come è stata esposta, richiede una notevole quantità di memoria, dato che per ogni combinazione bisogna memorizzare una stringa alfabetica.
Si è preferito perciò scrivere in memoria un vettore in cui ogni elemento indica, se marcato con 1, il *numero* della corrispondente combinazione valida, per risalire quindi dal numero della combinazione valida (ad esempio la 232esima) alla combinazione effettiva (cioè ad esempio Rosso, Verde, Giallo, Giallo). Questa soluzione risparmia memoria, ma rispetto alla soluzione "ideale" è più lenta.
L'ultimo commento riguarda la funzione BASIC MID (che compare ad esempio allo statement 230): questa funzione estrae da un insieme alfanumerico (primo vettore) un sottoinsieme alfanumerico, a partire da una posizione conosciuta di carattere (seconda variabile), e per un'ampiezza anch'essa nota (terza variabile).

■

```
10 REM
20 REM    MASTERMIND
30 REM
80 INPUT "NUMERO DI COLORI ";C9
90 IF C9>6 THEN PRINT " NON SUPERARE 6 !":GO TO 80
100 INPUT " NUMERO POSIZIONI ";P9
110 INPUT "NUMERO DI ROUNDS";R9
120 P=C9↑P9
130 PRINT " COMBINAZIONI COMPLESSIVE ";P
140 H=0:C=0
150 DIM Q(P9),S(10,2),S$(10),A$(P9),G$(P9),I(P),H$(P9)
160 L$="RGVBAM"
170 PRINT
180 PRINT
190 PRINT "COLORE          LETTERA INIZIALE"
200 PRINT "-------------------------------------------"
210 FOR X=1 TO C9
220 READ X$
230 PRINT X$;TAB(13);MID$(L$,X,1)
240 NEXT X
250 PRINT
260 FOR R=1 TO R9
270 PRINT
280 PRINT " NUMERO ROUND ";R
290 PRINT
300 PRINT " INDOVINA LA MIA COMBINAZIONE "
310 REM PROGRAMMA DI CODE-MAKER
320 A=INT(P*RND(1)+1)
330 GOSUB 3000
340 FOR X=1 TO A
350 GOSUB 3500
360 NEXT X
370 FOR M=1 TO 10
380 PRINT "MOSSA N. ";M;"LA DISPOSIZIONE E' ";:INPUT X$
390 IF X$="TAVOLA" THEN 2000
400 IF X$="QUIT" THEN 2500
410 IF LEN(X$)<>P9 THEN PRINT " NUMERO PEZZI ERRATO":GOTO 380
420 REM VERIFICA COLORE E IMPACCA X$ IN G$(1-P9)
430 FOR X=1 TO P9
440 FOR Y=1 TO C9
450 IF MID$(X$,X,1)=MID$(L$,X,1) THEN 480
460 NEXT Y
470 PRINT " <";MID$(X$,X,1);"> E' SCONOSCIUTO. ":GOTO 380
480 G$(X)=MID$(X$,X,1)
490 NEXT X
500 REM CONVERSIONE DI Q(1-P9) IN A$(1-P9)
510 GOSUB 4000
520 REM PER OTTENERE IL NUMERO DI BIANCHI E NERI
530 GOSUB 4500
540 IF B=P9 THEN 630
550 REM STAMPA RISULTATO
560 PRINT "HAI ";B;"NERI E ";W;"BIANCHI"
570 REM SALVATAGGIO MANO DI GIOCO PER EVENTUALE RICHIESTA
580 S$(M)=X$
590 S(M,1)=B
600 S(M,2)=W
610 NEXT M
620 PRINT "HAI SUPERATO IL NUMERO DI MOSSE":GOTO 640
622 GOSUB 4000
623 PRINT " LA COMBINAZIONE DI CODE-MAKER ERA: ";
624 FOR X =1 TO P9
625 PRINT A$(X);
626 NEXT X
627 PRINT
630 PRINT "HAI INDOVINATO IN ";M;"MOSSE!"
640 H=H+M
650 GOSUB 5000
660 REM
670 REM PROGRAMMA DI CODE-BREAKER
680 REM
690 FOR X=1 TO P
700 I(X)=1
710 NEXT X
720 PRINT "ORA INDOVINO IO. PENSA AD UNA COMBINAZIONE."
730 INPUT "PREMI IL TASTO RETURN QUANDO SEI PRONTO.";X$
740 FOR M=1 TO 10
750 GOSUB 3000
760 REM RICERCA DI UNA COMBINAZIONE
770 G=INT(P*RND(1)+1)
780 IF I(G)=1 THEN 890
790 FOR X=G TO P
800 IF I(X)=1 THEN 880
810 NEXT X
820 FOR X=1 TO G
830 IF I(X)=1 THEN 880
840 NEXT X
850 PRINT " NON DARMI INFORMAZIONI INCONSISTENTI."
860 PRINT " RIPROVIAMO E CERCA DI ESSERE PIU' ATTENTO."
870 GOTO 660
880 G=X
890 REM CONVERSIONE DEL NUMERO DI COMBINAZIONE G IN G$
900 FOR X=1 TO G
910 GOSUB 3500
920 NEXT X
930 GOSUB 6000
940 PRINT " LA MIA COMBINAZIONE E': ";
950 FOR X = 1 TO  P9
960 PRINT H$(X);
970 NEXT X
980 INPUT " NERI,BIANCHI ";B1,W1
990 IF B1=P9 THEN 1120
1000 GOSUB 3000
1010 FOR X=1 TO P
1020 GOSUB 3500
1030 IF I(X)=0 THEN 1070
1035 GOSUB 6500
1040 GOSUB 4000
1050 GOSUB 4500
1060 IF B1<>B OR W1<>W THEN I(X)=0
1070 NEXT X
1080 NEXT M
1090 PRINT " HO USATO TUTTE LE MIE MOSSE!"
1100 PRINT "CREDO CHE LA MIA CPU SIA IN FERIE"
1110 GOSUB 1130
1120 PRINT " HO UTILIZZATO ";M;"MOSSE!"
1130 C=C+M
1140 GOSUB 5000
1150 NEXT R
1160 PRINT " FINE GIOCO"
1170 PRINT " PUNTEGGIO FINALE :"
1180 GOSUB 5040
1190 STOP
2000 REM
2000 REM
2010 REM      ROUTINE ESECUZIONE RICHIESTA -TAVOLA-
2020 REM
2025 PRINT
2030 PRINT "   TAVOLA"
2040 PRINT "NUMERO MOSSA      COMBINAZIONE       NERI  BIANCHI"
2050 FOR Z=1 TO M-1
2060 PRINT Z;TAB(20);S$(Z);TAB(37);S(Z,1);TAB(43);S(Z,2)
2070 NEXT Z
2075 PRINT
2080 GOTO 380
2500 REM
2510 REM      ROUTINE DI QUIT
2520 REM
2530 PRINT " RINUNCI FACILMENTE! LA MIA COMBINAZIONE ERA :";
2535 GOSUB 4000
2540 FOR X=1 TO P9
2550 PRINT A$(X);
2560 NEXT X
2565 PRINT
2570 PRINT "ARRIVEDERCI"
2580 STOP
3000 REM
3010 REM INIZIALIZZA Q(1-P9) A ZERO
3020 REM
```

```
3030 FOR S=1 TO P9
3040 Q(S)=0
3050 NEXT S
3060 RETURN
3500 REM
3510 REM INCREMENTO DI Q(1-P9)
3520 REM
3522 IF Q(1)>0 THEN 3530
3524 REM SE ZERO, METTERE TUTTO A 1
3526 FOR S=1 TO P9
3527 Q(S)=1
3528 NEXT S
3529 RETURN
3530 Q=1
3540 Q(Q)=Q(Q)+1
3550 IF Q(Q)<=C9 THEN RETURN
3560 Q(Q)=1
3570 Q=Q+1
3580 GOTO 3540
4000 REM
4010 REM CONVERSIONE DI Q(1-P9) IN A$(1-P9)
4020 REM
4030 FOR S=1 TO P9
4040 A$(S)=MID$(L$,Q(S),1)
4050 NEXT S
4060 RETURN
4500 REM
4510 REM CALCOLO DEI NERI (B) E DEI BIANCHI (W)
4520 REM
4530 REM
4540 B=0:W=0:F=0
4550 FOR S=1 TO P9
4560 IF G$(S)<>A$(S) THEN 4620
4570 B=B+1
4580 G$(S)=CHR$(F)
4590 A$(S)=CHR$(F+1)
4600 F=F+2
4610 GOTO 4660
4620 FOR T=1 TO P9
4630 IF G$(S)<>A$(T) THEN 4650
4640 IF G$(T)=A$(T) THEN 4650
4645 W=W+1:A$(T)=CHR$(F):G$(S)=CHR$(F+1):F=F+2:GOTO 4660
4650 NEXT T
4660 NEXT S
4670 RETURN
5000 REM
5010 REM STAMPA PUNTEGGIO
5020 REM
5030 PRINT " PUNTEGGIO :"
5040 PRINT "              COMPUTER";C
5050 PRINT "              GIOCATORE";H
5060 PRINT
5070 RETURN
5500 REM
5510 REM CONVERSIONE DI Q(1-P9) IN G$(1-P9)
5520 REM
5530 FOR S=1 TO P9
5540 G$(S)=MID$(L$,Q(S),1)
5550 NEXT S
5560 RETURN
6000 REM
6010 REM CONVERSIONE DI Q(1-P9) IN H$(1-P9)
6020 REM
6030 FOR S=1 TO P9
6040 H$(S)=MID$(L$,Q(S),1)
6050 NEXT S
6060 RETURN
6500 REM
6510 REM COPIA DI H$ IN G$
6520 REM
6530 FOR S=1 TO P9
6540 G$(S)=H$(S)
6550 NEXT S
6560 RETURN
8000 REM DATI DEL PROGRAMMA = NOMI DEI COLORI
8010 DATA ROSSO,GIALLO,VERDE,BLU,ARANCIO,MARRONE
8020 REM
9999 END
```

BIAS
MICROELETTRONICA '80

Edizione BIAS
dedicata alla componentistica elettronica, ai minisistemi ed alla strumentazione di laboratorio.

- COMPONENTI ELETTRONICI DAGLI ELETTROMECCANICI PER APPLICAZIONI ELETTRONICHE, AGLI INTEGRATI VLSI.
- MICROCOMPUTER, MINISISTEMI, PERSONAL E HOME COMPUTER, PERIFERICHE OEM.
- APPARECCHIATURE E STRUMENTAZIONE PER PRODUZIONE, COLLAUDO, CERTIFICAZIONE E ACCERTAMENTO QUALITÀ NELL'INDUSTRIA ELETTRONICA.
- STRUMENTAZIONE DI LABORATORIO E PER LA RICERCA SCIENTIFICA ED APPLICATA.

Una opportunità unica in Italia per un contatto diretto con l'elettronica professionale.

Per informazioni e prenotazioni:
STUDIO BARBIERI
20129 Milano (Italia) - Viale Premuda, 2 - Tel. 796.096/421/635

in collaborazione con

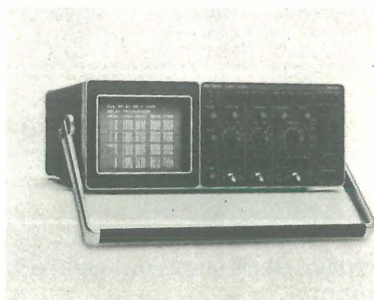
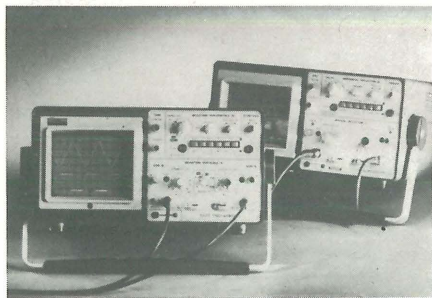
STRUMENTAZIONE

La Philips combina in unico strumento un oscilloscopio e un analizzatore logico

L'integrazione di un analizzatore a 10 MHz in un oscilloscpio da 25 MHz permette di realizzare un originale strumento per i servizi di manutenzione e prove. Il logiscopio PM 3540, presentato dalla Divisione Test & Measuring Instruments Philips, combina le prestazioni di un versatile oscilloscopio da 25 MHz con quelle di un analizzatore logico a 16 canali. Lo strumento può funzionare o come oscilloscopio, o come analizzatore logico, o svolgere entrambe le funzioni.
L'analizzatore logico permette di visualizzare, a scelta, sullo schermo dell'oscilloscopio, stati binari, ottali od esadecimali. Le caratteristiche principali sono: funzione di confronto, memoria di 64 parole con visualizzazione di 16 parole, qualificatori del clock e flessibili possibilità di triggerazione, compreso il ritardo; è possibile anche l'estensione del trigger mediante sonde aggiuntive.
Più in particolare, per quanto riguarda il trigger esistono, in questo strumento, ampie possibilità di trigger interno, esterno o manuale. La parola di trigger può venire rappresentata nel desiderato formato. La scelta del blocco di dati rappresentati sullo schermo si ottiene sia usando i controlli del cursore sia selezionando la condizione voluta dallo stesso flusso dei dati; ciò permette una rapida impaginazione dei dati.
Il campionamento è sincrono sino a 10 MHz. La soglia è variabile con posizione fissa a livello TTL. L'avviamento può essere manuale od automatico, con continua acquisizione e visualizzazione dei dati da parte dello strumento. Il trigger può essere ritardato fino a 9999 impulsi.
La memoria principale inoltre è costituita da 64 × 16 bits con visualizzazione sul tubo dell'oscilloscopio di 16 = 16 bits. Esiste la possibilità di concellazione di righe e di colonne. Il modo di funzionamento di confronto impiega due memorie separate da 64 parole. In una memoria può essere immagazzinata una tabella di stati per il confronto coi nuovi dati raccolti. Eventuali diseguaglianze sono indicate in testata e marcate con digits intensificati.
L'analizzatore logico ha 16 canali d'ingresso mediante tre gruppi da otto sonde miniatura, che consentono anche l'ingresso per il clock esterno e per tre qualificatori del clock.
La sezione oscilloscopio consiste in un sofisticato strumento per manutenzione con sensibilità di 2 mV sull'intera larghezza di banda di 25 MHz. Esso è provvisto di due ingressi per i canali verticali completamente separati da quelli dell'analizzatore logico.
In questo strumento esiste un'ampia scelta di possibilità di trigger che comprende il trigger automatico sul picco-picco e l'accoppiamento in continua. Le sorgenti di trigger possono essere l'ingresso verticale, il composto, l'esterno o l'analizzatore logico.
Grande attenzione è stata prestata al progetto ergonomico per essere sicuri che entrambe le parti dello strumento fossero semplici e facili da usare. Uno schermo largo 80 × 100 mm assicura una chiara visualizzazione delle presentazioni sia dell'oscilloscopio che dell'analizzatore logico.
Prove di accensione e routines di di manutenzione incorporate assicurano la massima affidabilità dello strumento e riducono al minimo il tempo morto. Il PM 3540 può funzionare con una grande varietà di alimentazioni, cioé: 110, 127, 220 e 240 V ac e da 21 a 27 V dc. Le sue dimensioni sono: 150 × 330 × 445 mm e l'indicatore logico pesa soltanto 8 kg.

PHILIPS-MONZA

Oscilloscopi economici - 15 MHz 2 canali - X - Y

La Enertec-Schlumberger presenta due oscilloscopi a 2 canali 5 mV/div., specifici per i settori industria, manutenzione e didattica.
Il modello 5023 per impieghi universali è caratterizzato da un CRT 8 × 10 cm, base tempi fino 200 ns, funzionamento XY, sincronismo TV, somma algebrica, vernieri di regolazione ecc.
Il modello 5013 è identico al 5023, ha solo in meno i vernieri dei canali Y e la possibilità della somma algebrica.
Compatti e luminosi, si portano

agevolmente grazie alle piccole dimensioni ed al peso ridotto e possono operare sia orizzontalmente che verticalmente.
Vengono assemblati con i più aggiornati criteri della tecnica: inserzione automatica dei componenti, test automatici al 100% delle piastre a mezzo di sistemi computerizzati per la più alta affidabilità.
L'uso di componenti standards assicura, anche a lunga distanza, costi di manutenzine ridotti, mentre il costo contenuto garantisce all'acquirente un rapporto prestazioni-/prezzo conveniente.

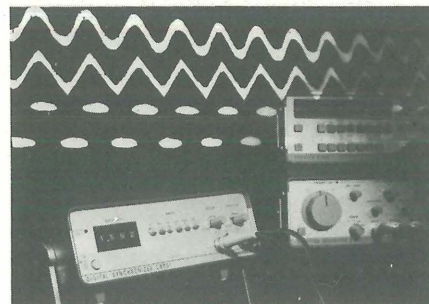
dB - ELECTRONIC INSTRUMENTS

Tester per piastre con microprocessori

La Olivetti Tecnost presenta l'apparecchiatura T80B per il collaudo di pastre dotate di microprocessori e componenti LSI.
Il T80B consente di eseguire dei tests sulla piastra in prova, alla velocità effettiva di funzionamento, realizzando perciò un'interfaccia di prova equivalente a quella reale.
Le famiglie di microprocessori per cui è già disponibile il software di collaudo sono: 8080-8085-8048-Z80-8021-F8. Sono in corso ulteriori ampliamenti per abbracciare altre famiglie di microprocessori.
Questa apparecchiatura, concepita e realizzata per il collaudo di piastre dotate di microprocessore, può ugualmente essere utilizzata per il collaudo di piastre che ne sono prive, nel qual caso sostituisce al microprocessore mancante la propria unità centrale.
Il T80B realizza un nuovo approccio di collaudo e di diagnosi dei difetti legato alla capacità autodiagnostica intrinseca del microprocessore in prova. Questo consente di contenere il costo dell'apparecchiatura conferendole perciò un alto grado di competitività nella risoluzione dei problemi di collaudo di unità di controllo, governi con microprocessori, anche nel caso di produzioni non professionali.
Il programma di test è residente su EPROM montate su di una cartuccia estraibile, e questo per rendere agevole l'adattamento del T80B al collaudo di differenti tipi di piastre.

OLIVETTI TECNOST - IVREA

Sincronizzatore digitale per generatori

Il sincronizzatore C 9001 della Siemens consente di rilevare in forma digitale e con elevata precisione (quarzo) le frequenze di un qualsiasi generatore di funzioni provvisto di presa per il comando esterno delle frequenze (ingresso VCO).
La gamma di frequenza può essere predisposta su sei decadi, da 10 Hz a 10 MHz, mediante un selettore a tamburo a quattro cifre. Le eventuali correzioni (taratura successiva) per il livello del multivibratore e per l'ampiezza di uscita vengono segnalati da un diodo luminoso.
Le prestazioni dei generatori di funzioni per quanto riguarda la regolazione e la deriva delle frequenze sono inferiori a quelle degli oscillatori a quarzo; spesso anche la riproducibilità della frequenza d'uscita è insoddisfacente. Il sincronizzatore digitale C 9001, accoppiato ad un generatore di funzioni, elimina questi inconvenienti.
Il principio di funzionamento può essere paragonato con quello di un comparatore di frequenza. Il sincronizzatore digitale preleva dal generatore di funzioni la frequenza intrinseca (frequenza istantanea) e la confronta con quella propria di riferimento regolata in forma digitale. La tensione pilota proporzionale di segno esatto, derivata dalla differenza tra le due frequenze, regola la frequenza intrinseca del generatore di funzioni sul valore di riferimento.

L'apparecchio, tenendo conto del campo di misura prescelto, presenta una risoluzione di frequenza dello 0,01% ed una deriva di 10^{-6}/°C.

SIEMENS ELETTRA - MILANO

DPM miniaturizzato

Lo strumento digitale da pannello VN 20 della AOIP, grazie alle sue piccole dimensioni, può essere facilmente inserito nei sistemi industriali.
Dotato di 4 ranges per la misura di tensioni c.c., il VN20 è essenzialmente caratterizzato da 20.000 punti di misura, ha un display a LED a 7 segmenti con 5 cifre alte 8 mm, la polarità e l'azzeramento sono automatici.
Lo strumento funziona con una tensione di alimentazione esterna di 5V, con un assorbimento di 350 mA.
La temperatura di funzionamento può variare da 5°C a 40°C, con un coefficiente di temperatura dello zero minore di 0,1 d/°C e un coefficiente di temperatura della sensibilità minore di $25 \cdot 10^{-6}$/°C da 200 mV a 2V e di $75 \cdot 10^{-6}$/°C da 20 a 200V.
Il DPM, che effettua 3 misure al secondo, può sopportare una tensione di 250Vrms o 300V picco, presenta un tempo di integrazione del segnale di ingresso di 80 ms e un settling time di 2s.
Nei 4 ranges di misura di 200mV, 2000mV, 20V e 200V in continua, la risoluzione e l'impedenza di ingresso sono rispettivamente: 10µV - 1000MΩ, 100µV - 1000MΩ, 1mV #, 1MΩ, 10mV # 10MΩ, mentre la precisione è in tutti i ranges pari a ± 0,02% della lettura e ± 0,01% del fondo scala.
Le dimensioni di questo DPM sono 25 × 75 mm per il pannello frontale e 117 mm di profondità.

AOIP

MEMORIE

Componenti Siemens per microprocessori

I componenti SAB 8155 e SAB 8156, una combinazione di memorie I/O e contatori sono disponibili con tempi di accesso di 140 ns; le due memorie RAM SAB 2114 (1K × 4) e SAB 2147 (4K × 1) possono venire collegate al SAB 8085 senza frapporre cicli di attesa. Vi è inoltre la RAM 8185 (1K × 8) con ingresso multiplex.
Nel campo delle memorie standard, il progresso tecnologico ha portato dei netti miglioramenti per quel che riguarda capacità, tempi di accesso e potenza dissipata. Infatti, accanto alle EPROM SAB 2716 (2K × 8) e SAB 2758 (1K × 8), troviamo la nuova SAB 2732, una memoria con una capacità di 4K × 8 bit con tensione di alimentazione a +5V. Uno speciale modo di funzionamento, che tiene conto dei componenti non attivati, consente infine di realizzare una maggiore densità di impaccamento sui singoli moduli con una potenza dissipata minore.
La Siemens, con le sue nuove memorie ROM, ha fatto ancora un passo avanti: alle EPROM corrispondono, con compatibilità dei pin, le ROM SAB 2316 (2K × 8) e SAB 2332 (4K × 8), mentre la SAB 2364 arriva fino a 8K × 8. Questo componente è alloggiato in una custodia con 28 pin e con un assegnamento dei segnali che lo rensono parzialmente compatibile con le altre ROM ed EPROM. Ciò semplifica il passaggio da un sistema all'altro.
Oltre a questi componenti per il microprocessore SAB 8085, vi sono anche buoni componenti di controllo per la periferia più esterna.
Le precedenti serie SAB 825 × e SAB 827 × per la trasmissione in serie di dati, per comando CRT ecc. vengono ora integrate dalla serie SAB 829 ×.
Ne fanno parte i due componenti SAB 8291 e SAB 8292 per l'allacciamento di apparecchi di misura ad un sistema di microcomputer tramite un apposito bus secondo le IEEE 488.
La SAB 8295 serve al collegamento di una stampante e matrice di punti 5 × 7.

SIEMENS ELETTRA-MILANO

RAM non volatile da 1024 bit

La M 120 della SGS-ATES è una memoria non volatile che l'utente può considerare come una RAM con un tempo di accesso rapido e un crite cycle molto più lento. È strutturata in un'organizzazione a 256 × 4 completamente decodificata, e può funzionare nei modi read e modify. Il modo modify effettua simultaneamente la scrittura e la cancellazione sulla parola dell'indirizzo.
Altre caratteristiche salienti della M120 sono: input latches per gli indirizzi, chip select e data in; dati di uscita latched e validi per il ciclo successivo; tempo di accesso di 450 ns per l'M120-2 e di 700 ns per l'M120; tempo di modifica della parola minore di 100 ms (la fine dell'operazione di modifica è indicata da un flag); 10.000 cicli di modifica per parola.
Queste memorie sono costruite con la tecnologia N-channel, Si-gate, double poly-silicon MOS e hanno uscite a drain aperto TTL-compatibili.
Necessitano delle alimentazioni V_{DD}= 12 ±10% e V_{pp} = 25V ±5%, con un consumo di 300 mW che si riduce a 100 mW in standby.
Il range di temperatura di funzionamento va da 0 a 70°C.
Le M120 e M120-2 vengono fornite in package dual-in-line ceramico a 18 pin.

SGS-ATES-AGRATE BRIANZA (MI)

RAM statica da 4K TTL compatibile per memorie ad alta velocità

La AMI Microsystems ha annunciato una nuova versione ad alta velocità della RAM Statica S2114, che aveva un tempo di accesso massimo di 150 ns o più.
Questa nuova versione, in tecnologia VMOS, che si chiama S2114H, ha un tempo di accesso massimo di 70ns. Il suo uso è particolarmente indicato per quelle applicazioni ad alta velocità nelle quali non sono usabili i componenti standard; la sua organizzazione di 1K × 4 e la singola tensione di alimentazione la rendono particolarmente adatta a questi sistemi.
La RAM Statica S2114H è un dispositivo da 4096 bit, in tecnologia VMOS, alimentato con una singola alimentazione da 5V; è completamente TTL-compatibile sia in ingresso che in uscita e pin di I/O comuni la rendono particolarmente adatta ai sistemi con bus dati bidirezionali.
La S2114H non ha bisogno di alcun ciclo di rinfresco. La funzione di chip select facilita l'espansione dei sistemi di memoria permettendo di collegare insieme i pin di I/O di diversi dispositivi.
La tecnologia ad alta velocità VMOS è un brevetto AMI e permette la costruzione in larga scala di dispositivi MOS/LSI ad alte prestazioni, con circuiti molto complicati.

AMI MICROSYSTEMS-MILANO

EPROM ad alimentazione singola

La Texas Instruments ha annunciato la messa in produzione di una EPROM monoalimentazione, la TMS 2508JL.
Si tratta di una 8K EPROM, pin-compatibile con la TMS 2516JL e la TMS 2532JL, disponibile in due range di velocità (−25 e −30); la memoria ha il chip select/power down automatico ed è, come le altre della famiglia, completamente statica.
Questa EPROM non è un prodotto derivato dalla 16K EPROM, come la 2758 INTEL, che è un fall-out della INTEL 2716; lo si vede anche dal tempo di accesso che si tratta di un prodotto disegnato a parte con tecniche moderne: fino ad oggi infatti non esistevano EPROM con tempo di accesso inferiore ai 350 ns.

TEXAS INSTRUMENTS-RIETI

SISTEMI DI SVILUPPO

Sistema di sviluppo STARPLEX

La National Semiconductor ha presentato una versione avanzata del proprio sistema di sviluppo STARPLEX, caratterizzata da un modulo di elevate prestazioni per l'emulazione interna al sistema (ISE) e da una serie di nuove opzioni di prezzo vantaggioso. Le opzioni comprendono dei potenziamenti di software, la linea di trasmissione STARLINK ed una stampante da 120 caratteri/s.

Il sistema STARPLEX apporta una semplicità operativa allo sviluppo di sistemi a microprocessore e delle caratteristiche estese per il collaudo, l'analisi e la correzione hardware e software dei prototipi. Il modulo ISE supporta l'8080 e supporterà le famiglie di microprocessori 8070 e 8048.
Il modulo ISE permette l'emulazione in tempo reale di un microprocessore e l'emulazione contemporanea di due microprocessori. Altri vantaggi sono il costo ridotto di un modulo ISE generalizzato per il supporto di diversi microprocessori da valutare e l'accesso completo alle strutture del sistema STARPLEX. Le elevate prestazioni sono ottenute mediante una CPU di ISE indipendente, una memoria a mappa dedicata e buffer di elevata velocità collocati vicino al microprocessore in esame.
Un sistema STARPLEX standard consiste nella CPU, in 64K bytes di memoria, in duplici unità a disco flessibile standard da 8" (che possono essere portate a quattro mettendo a disposizione 1 M byte di memoria on-line), un video di controllo, una tastiera ed una stampante da 50 caratteri/s.
Un sistema standard comprende anche il seguente software: il sistema operativo e l'editor adattati al terminale video ad alta velocità, che forniscono delle funzioni estese al singolo abbassamento di tasto; FORTRAN e BASIC; uno speciale linguaggio interattivo di prova; assembler e macroassembler; programmi di utilità ed un package diagnostico per il sistema. Vi sono dei cross-assemblers con capacità macro per gli 8080, 8048/49/50 e 8070 e un debugger simbolico ed assembler e disassembler in linea per permettere di effettuare delle modifiche immediate durante le fasi di collaudo e di debugging.

NATIONAL SEMICONDUCTOR-MILANO

Nuovo sistema di sviluppo HP per lo sviluppo di prodotti a microprocessori

Il nuovo Sistema di Sviluppo Logico della Hewlett-Packard, denominato modello 64000, ha una architettura a più stazioni di lavoro e adotta un disco rigido per ottenere un sistema operativo flessibile di elevate prestazioni. Questa appa-

recchiatura è di ausilio ai progettisti software ed hardware nelle fasi di progetto, messa a punto e ricerca guasti dei prodotti a microprocessore.
Il modello 64000 svolge le tipiche funzioni di sviluppo della programmazione (quali l'editing, la gestione dei files, l'assemblaggio, la compilazione ed il linking) ed inoltre effettua l'emulazione in tempo reale dei più diffusi microprocessori, ed è dotato di un analizzatore logico in tempo reale.
Attualmente, quattro sono i microprocessori (Intel 8080 e 8085, Motorola 6800 e Zilog Z80) che l'apparecchiatura assiste in modo completo con macroassemblers rilocabili e con emulatori in tempo reale, alla velocità dei microprocessori.
La moderna archittetura del Sistema di sviluppo è stata progettata in modo da consentire future espansioni (microprocessori a 8,16 e 32 bit).
Caratteristiche importanti del Sistema di Sviluppo Logico sono:
- Memoria di massa su disco rigido da 20 a 120 Megabytes
- Architettura multistazione, che va da un sistema minimo costituito da una stazione di sviluppo, disco e stampante, fino al sistema a sei stazioni di sviluppo funzionanti simultaneamente, e utilizzanti in comune il disco e la stampante.
- Elevata interattività con l'operatore mediante controllo a cursore e Direct Syntax (sintassi guidata) sul visore mediante otto tasti di funzione.
- Moduli di emulazione dedicati ai vari microprocessori con macroassemblers rilocabili, emulazione in tempo reale, e analisi trasparente in tempo reale.

Il sistema base consiste di una stazione di sviluppo modello 64100 A, un disco rigido HP e una stampante. Ogni stazione di sviluppo è dotata di un processore ospite (host-processor) con 64 K di memoria, di un controllo di I/O e di un controllo del display. Ogni stazione accetta, facilmente accessibili, dieci schede per le opzioni del sistema.
Il sistema ospitante e il sistema emulatore hanno bus e memorie separati, evitando così tutti i problemi di conflitto di accesso ai bus.
Ciascuna stazione di sviluppo può essere configurata in modo adatto alle applicazioni aggiungendo opzioni. Attualmente queste opzioni comprendono: emulatori in tempo reale per quattro importanti microprocessori, fino a 128 kilobytes di memoria di emulazione da 200 nanosecondi, un drive di nastro magnetico a cartuccia per salvataggio di files e *software entry*, un programmatore di PROM completo, un modulo analizzatore logico in tempo reale, e un'interfaccia seriale RS-232-C (V24). La HP sta implementando, a breve termine, il sistema con un compilatore Pascal.
L'auto-test è incluso. Mediante impiego di "signature analysis", documentazione e "troubleshooting trees" viene resa pratica e possibile l'assistenza sul posto.
Il grande volume di memoria di massa e la rapida velocità di risposta del disco rigido rendono possibile un sistema operativo veramen te sofisticato. Gli errori sintattici sono virtualmente eliminati mediante "Directed Syntax" e otto tasti di funzione.
Le funzioni comandate da questi tasti variano da operazione a operazione, ma quelle attivabili di volta in volta sono visibili sopra il tasto nel display, e quindi l'operatore può evitare di ricorrere continuamente al manuale d'uso. Separate funzioni di controllo dei testi consentono grande flessibilità nella gestione di testi di qualsiasi lunchezza.
Secondo l'HP, con il sistema 64000 si può facilmente e velocemente mettere a punto i programmi e trovare gli errori circuitali. Una base di dati condivisa permette l'accesso alle copie correnti di ogni programma in qualsiasi momento. Con una sola unità, la stazione di sviluppo consente di scrivere i programmi, emulare i circuiti e attuare le funzioni di un analizzatore di stati logici.
I moduli di emulazione, dedicati a diversi microprocessori, consentono all'utente di acquistare un solo sistema di sviluppo per ogni particolare microprocessore, sia a 8 che a 16 Bit.

HEWLETT-PACKARD

ICE-49: nuovo emulatore circuitale per la famiglia MCS-48

L'ICE-49 è l'emulatore circuitale che permette di interfacciare i sistemi di sviluppo Intel con qualsiasi prototipo realizzato con la famiglia MCS-48.
La famiglia MCS-48 comprende i microcomputers 8049, 8048, 8748, 8039, 8035 e 8021.
L'ICE-49 è un modulo che risiede nel sistema di sviluppo Intellec e si interfaccia con il microcomputer della famiglia MCS-48 tramite un apposito cavo. Con l'emulatore ICE 49 il progettista ha la possibilità di far funzionare il suo sistema in tempo reale e di tracciare, sempre in tempo reale, sino a 250 cicli di istruzioni.
In aggiunta il progettista può far lavorare il sistema passo-passo per controllare più da vicino l'esecuzione del programma. Con la memoria statica di cui è provvisto l'ICE-49 si può emulare il programma del prototipo realizzato con l'MCS-48.
Il progettista può anche visualizzare e alterare dati di memoria, registri, flags e porte di I/O.
Notevoli sono le possibilità di debugging dell'ICE-49: la possibilità di far riferimenti simbolici permette al progettista di usare simboli anzichè valori assoluti tutte le volte che occorre modificare ed esaminare sezioni del prototipo.

ELEDRA - MILANO

Modulo di valutazione per MC6801

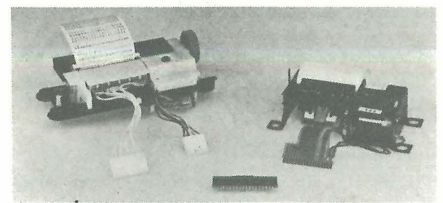
La Motorola ha preparato un modulo di valutazione per microcomputers, MEX6801EVM, il quale mette a disposizione degli utenti un ambiente appropriato per la valutazione del sofisticato microcomputer MC6801.
Il modulo costituisce un sistema completo hardware e software su una scheda singola di basso costo, che può essere inserita in un EXOR ciser od usata come sistema indipendente.
La trasmissione dati avviene attraverso la porta seriale usando un terminale compatibile RS-232C.
Il modo di funzionamento del microcomputer MC6801 è selezionabile dall'utente per mezzo di collegamenti a ponticello sulla scheda. Nella configurazione modulare più semplice, l'MC6801 può funzionare in modo Chip Singolo, usando il monitor di messa a punto LILbug che è programmato per mascheratura sul chip MC6801.
La scheda contiene l'oscillatore e l'interfaccia RS-232C, che sono gli unici dispositivi di supporto necessari intorno al MC6801. Nella RAM su chip rimane dello spazio sufficiente per un semplice programma di I/O dell'utente, e la tabella di I/O programmato normalmente, stabilita nella RAM del LILbug, può essere facilmente modificata dall'utente per scopi di valutazione.
È previsto dello spazio sulla scheda anche per 4K byte ulteriori di RAM, 2K byte di Earasable Programmable ROM, un Timer programmabile ed un ACIA, consentendo così di assemblare dei sistemi complessi, definiti dall'utente, in un modulo singolo quando l'MC6801 funziona nel Modo Espanso. Inoltre, è incorporata la decodifica d'indirizzo per 4K byte di memoria esterna alla scheda.
Per consentire la costruzione di interfacce particolari o di altri dispositivi, è compresa una vasta zona di cablaggio, per collegamenti wrap-pati. Le specifiche di alimentazione per la scheda completa sono + 5V a 1A e ± 12V a 50 mA.

MOTOROLA - MILANO

PERIFERICHE

Nuove stampanti termiche Olivetti

Sono prodotti dalla Olivetti nuovi tipi di stampanti termiche che si affiancano alla usatissima PU 1800. Esse sono la PU 1808, la PU 1828 e la PU 1840.

- La PU 1808 è caratterizzata dalla durata di un milione di linee e dal basso consumo, tale da renderla utilizzabile nelle applicazioni con batteria. Ha 16 posizioni di stampa per linea ed una velocità di due linee al secondo.
- La PU 1828 differisce dalla precedente per le posizioni di stampa per linea, che sono 28, e per la notevole velocità: 4 linee al secondo. Questa velocità, la più alta di tutta la serie, viene ottenuta raddoppiando il numero degli elementi termici che si muovono per formare il carattere di stampa.

- La PU 1840 ha 40 posizioni di stampa per linea e velocità di 4/2 righe al secondo. Può essere gestita da un controllore PU 41 opportunamente programmato, che può essere collegato ad un sistema in due modi:
  a) con interfaccia parallela; e quindi direttamente connesso sul bus di sistema o attraverso le porte di Input/Output;
  b) con Remote Printer Controller su interfaccia seriale RS 232C. La stampa viene attivata automaticamente quando la riga è completa oppure al caricamento di caratteri speciali (Line Feed - Carriage e Return).

Per la loro struttura compatta e completa le stampanti sono in grado di soddisfare le esigenze di stampa nelle calcolatrici, nelle bilance elettroniche, nei minicomputers, nei registratori di casse, e in svariatissime altre applicazioni.

ELEDRA-MILANO

Allo Smau novità della Honeywell I.S.I. nel campo delle stampanti

La Honeywell Information Systems Italia ha presentato quest'anno allo SMAU due nuove stampanti a matrice, particolarmente indicate per il mercato OEM, che si pongono ai due estremi della famiglia Serial Printer Product Line.
Si tratta rispettivamente della R 28 e della S10.
La prima, una 132 colonne con stampa a 180 cps bidirezionale ottimizzata, offre una vasta gamma di prestazioni e funzionalità: oltre alla stampa compressa, ai caratteri di doppie altezze o larghezza, ai macrocaratteri, alla sottolineatura, possiede una ricca possibilità di tabulazioni, un doppio insieme di caratteri, la memoria non volatile ed una sofisticata gestione del buffer.
L'interfaccia è seriale, la velocità di trasmissione arriva fino a 9600 bps con "overlapping" di stampa e tra-

smissione fino a 2400 bps. Tutte e opzioni della famiglia sono applicabili su questa macchina, in particolare il trascinamento indipendente dei moduli.
Si tratta, in definitiva, di una teleprinter di una versatilità notevole, in grado di soddisfare le diverse esigenze e di porsi, grazie alla serietà del suo approccio tecnologico, come elemento interessante e di alta qualità nel campo delle stampanti.
L'altra, la S 10, si è subito delineata come un prodotto di "largo consumo".
Indirizzata al mercato dei mini, micro e personal computers, presenta caratteristiche salienti per quanto riguarda la semplicità di concezione, in particolare della sua meccanica.
È una 80 colonne, 80 cps bidirezionale, a matrice, con interfaccia seriale asincrona, 1200/9600 bps di trasmissione, buffer di linea, motori a step e tear-bar per uno strappo veloce ed economico della carta (a 4 cm dalla riga di stampa).
Il suo basso costo, la sua elettronica tutta su un'unica scheda, l'assenza di problemi di manutenzione preventiva sono già traguardi significativi; inoltre il design lineare e piacevole e le dimensioni ridotte al minimo (17 × 41 × 32 cm) ne completano le caratteristiche di compattezza e semplicità.
Un ultimo dato interessante: entrambe le macchine sono interamente progettate e costruite in Italia.

HONEYWELL I.S.I-MILANO

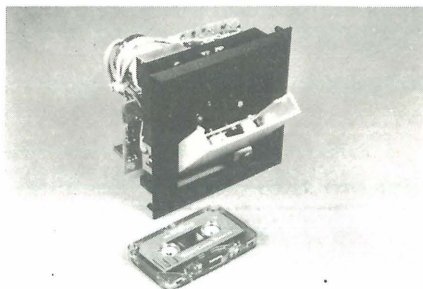
Nuova stampante ad impatto a basso costo Mactronics

"Tigre di carta" è la novità nel campo delle stampanti ad impatto per soddisfare le esigenze relative al campo dei minisistemi ad un costo estremamente ridotto.
Queste le caratteristiche standard:
- Set di 96 caratteri ASCII (upper e lower case e caratteri speciali).
- Grandezza dei caratteri selezionabile software (8 possibili grandezze.)
- Possibilità di uso di carta a più copie.
- Forms length control.
- Interfaccia parallela e seriale (Centronics-compatibile e RS 232C).
- Multiple line buffer.
- Formato di 80 e 132 caratteri.
- Velocità massima di stampa 198 caratteri al secondo.
- Self-test automatico all'accensione.

Opzionale la possibilità di Full dot Graphics.

MACTRONICS ITALIA-MILANO

Mini-cassette trasport della Braemar

La Microlem Data distribuisce il "CASSETTE TRANSPORT" CM-600 della Braemar nelle diverse versioni: 2400 baud, 4800 baud e Read While Write.
Progettato per applicazioni digitali, il CM-600 misura soltanto 76 × 76 × 64 mm. e pesa solo 230 gr.
Comprende tutta la logica di controllo ed anche gli amplificatori di lettura/scrittura che lo rendono complementare TTL-compatibile.
Richiede una sola alimentazione (+ 5 VDC) e in Read/Write consuma meno di 1 Watt.

Nonostante le ridotte dimensioni, ha una capacità di 160 KBytes su minicassette da 80 ft. (sono disponibili anche minicassette da 50 ft.).
Tutto questo fa del CM-600 un'unità veramente pronta per l'uso in applicazioni digitali dove semplicità di colloquio, basso costo e piccole dimensioni sono fattori fondamentali.

MICROLEM DATA-Vimercate (MI)

Lettore di schede perforate o marcate per gli utenti dei sistemi serie 1 IBM

La Mactronics Italia, nell'intento di aumentare le possibilità di utilizzo del sistema serie 1 IBM, ha sviluppato per lo stesso una gamma di prodotti compatibili tra i quali un sottosistema di lettura di schede.
L'unità è composta da lettore di schede Documation (da 300 sk al minuto) e da un'interfaccia con uscita seriale RS-232 con possibilità di selezionare la velocità di comunicazione da 110 a 9600 Bit al secondo.
Il dispositivo viene collegato al TTY Adapter della Serie 1 e consente a 9600 bps una lettura di 205 schede al minuto.

MACTRONICS ITALIA-MILANO

PERSONAL COMPUTERS

La Rockwell presenta il microcomputer AIM 65 a memoria a bolla con capacità di un milione di bit

Un microcomputer AIM 65, nella versione di 1K byte RAM con sistema di memoria a bolla pienamente indirizzato e estensione di un megabit, è stato presentato all'ELECTRO 79 dalla Rockwell International.
La chiave di accesso per la memoria esterna con l'AIM 65 è un motherboard come sviluppo di memoria di transito, che la Rockwell ha definito con la dicitura "AIM 65 Expansion Motherboard" e che ora è offerto come opzione al microcomputer R6500.
Originariamente progettato come microprocessore ad aiuto educativo, l'AIM 65 è stato adottato da un migliaio dei suoi primi compratori per applicazioni come sistema di sviluppo del prodotto R6500, usato come terminale e utilizzato per una vasta gamma di processi di controllo. Durante l'Electro '79 la Rockwell ha dimostrato il microcomputer AIM 65 in tutte queste sue applicazioni.
Lo sviluppo motherboard dell'AIM 65 ha cinque canali di collegamento nel suo connettore e può essere adottato dai modulatori del SY STEM 65 della Rockwell o da quelli della Motorola Exorcisor, così come per i modulatori aggiunti da Burr Brown o da altri costruttori.
Essenzialmente il motherboard estende l'AIM 65 in linea. Le "lineebus" del sistema (indirizzamento, dati e controllo) sono bufferizzate per provvedere ampiamente alla capacità di indirizzamento. La de-

codifica di indirizzamento logico per il controllo interno ed esterno è realizzata ad indirizzi con incrementi da 4K bytes. Sedici switches ne permettono l'utilizzo per definire ogni porzione di 4K bytes nell'area di indirizzamento del R 6502 di 65K bytes sia interno che esterno all'AIM 65.
Il dimostratore AIM 65 è stato progettato per indirizzare, sotto controllo software, i 128.000 byte di capacità del modulo di memoria a bolle della Rockwell. La memoria a bolle è usata come file-archivio permanentemente. Un nastro magnetico può essere impiegato similarmente con velocità operative notevolmente inferiori.
Un altro test per l'AIM 65 è stata una elaborazione con terminale periferico.
A questo scopo il microcomputer è stato collegato con un modulatore Rockwell R24 2400-bps, e i segnali numerici sono stati inviati tramite un conduttore elettrico ordinario da un terminale SOROC 10120.
Un terzo test per l'AIM 65 Rockwell è stata una applicazioni di progettazione per un impianto chimico. Questo test utilizza un 4K AIM 65 che controlla tre serbatoi di attesa, uno di miscela, quattro di stoccaggio ed il loro scarico in un serbatoio di trasporto.

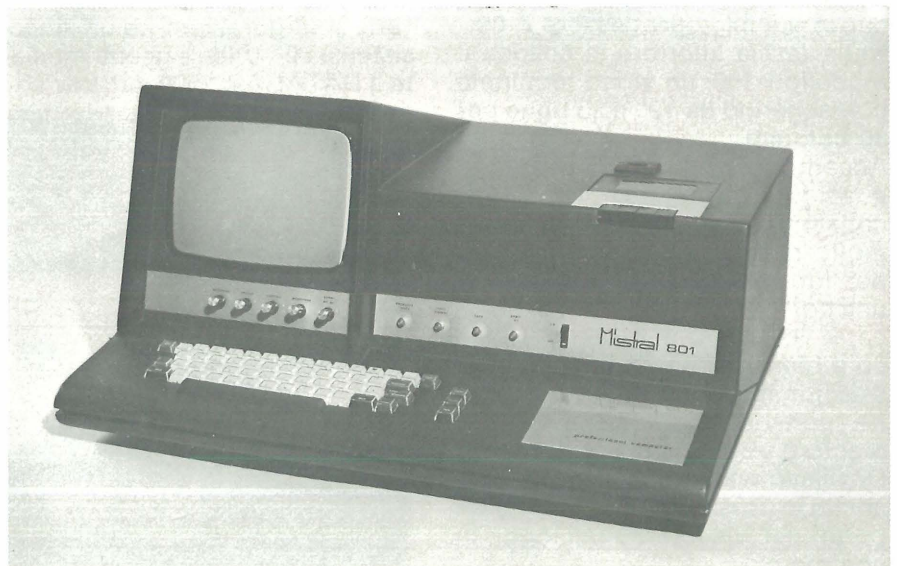
DE MICO - MILANO

Personal computer PC 100

La Siemens ha messo a punto un sistema a microcomputer che può essere utilizzato come calcolatore da tavolo liberamente programmabile oppure come un'agenda elettronica per appunti.
IL PC 100, pur rispondendo alla diffusa richiesta di calcolatori per uso industriale, è pensato soprattutto per applicazioni nel campo professionale. Con la sua tastiera da macchina per scrivere ed un visualizzatore alfanumerico a 20 caratteri, questo personal computer viene a colmare un vuoto tra i semplici apparecchi di base, come per esempio il Mikroset 8080, ed i costosi modelli dotati di monitor. È previsto in un secondo tempo il collegamento a un videoterminale.
Il PC 100 può essere fornito in forma di modulo equipaggiato e collaudato, oppure come apparecchio a sé stante, pronto per essere collegato alla rete.
Con il governo centrale del PC 100 si possono programmare o automatizzare numerosi processi meccanici, elettrici, termici o chimici, oppure rilevare, in base ad un programma, dati statistici riguardanti, ad esempio, la produzione e lo sviluppo; inoltre un PC 100, equipaggiato con uno speciale accessorio, è in grado di controllare fino a 15 apparecchi di misura per laboratori, per sale prova o per controlli di qualità.
Utilizzando il linguaggio di programmazione BASIC, questo personal computer diventa un calcolatore da tavolo, capace di risolvere anche problemi matematico-scientifici. Infine, chiunque è in grado di programmare il PC 100 come promemoria per numeri telefonici, indirizzi o scadenze. Può essere utilizzato perfino come ausilio didattico per lo studio di lingue straniere, introducendo e leggendo poi i dati, secondo determinati criteri.
Per usi professionali, il personal computer viene fornito come modulo aperto - PC 100 KIT - completo di tastiera ASCII, di visualizzatore alfanumerico LED a 20 caratteri e di stampante termica a 20 caratteri (120 righe al minuto). La memoria di lavoro è costituita da una RAM a 1K byte, mentre un programma di monitor ad 8K bytes offre numerose possibilità di controllo. Il bus di sistema fa capo ad un connettore che consente collegamenti per eventuali ampliamenti. Vi sono poi altri punti di collegamento per telescrivente, registratore a cassette e canali I/O.
L'apparecchio PC 100 finito può essere impiegato subito dopo l'allacciamento alla rete. Esso dispone di una capacità di memoria quadrupla (RAM 4K bytes) e di un BASIC a 8K bytes (ROM).
A parte queste differenze, i dati tecnici sono i medesimi per entrambe le versioni, per le quali vengono preparati attualmente moduli interfaccia verso videoterminale che consentono una rappresentazione grafica sullo schermo in bianco e nero, utilizzabile per giochi, passatempi e istruzione.

SIEMENS ELETTRA - MILANO

Un personal computer prodotto in Italia

Il Mistral 801 è un Personal Computer prodotto interamente in Italia dalla Mistral S.P.A. di Sermoneta (Latina) su licenza della APF Electronics Inc. New York.
Il sistema è costituito attorno al microprocessore 6502 e ha una struttura modulare basata su cinque moduli diversi. Questi moduli sono i seguenti: modulo CPU, modulo memoria ROM, modulo interfaccia cassetta e porta seriale, modulo interfaccia terminale video e uno o due moduli memoria RAM ciascuno da 16 kbytes rispettivamente per la versione a 16 kbytes e 32 kbytes.
Il calcolatore si presenta in forma compatta contenendo nello stesso involucro il terminale video a 9" da 16 righe e 40 caratteri per riga, un registratore a cassetta con capacità di memorizzazione di 80 kbytes,

la tastiera oltre al cestello con i moduli.
Il linguaggio contenuto nel Mistral 801 è derivato dal Joss, sviluppato dalla Rand Corporation. Studiato particolarmente per ridurre al minimo i comandi questo linguaggio è orientato verso l'utente non esperto in programmazione pur rimanendo un linguaggio molto potente.
La caratteristica più saliente è comunque la possibilità di effettuare una programmazione strutturata. Sia il sistema operativo che il linguaggio sono contenuti su 24 Kbytes di ROM. Le caratteristiche del Mistral 801 lo rendono adatto per il campo ingegneristico essendo il linguaggio efficace per il calcolo matriciale. Altre applicazioni altrettanto efficaci sono: calcoli statistici e finanziari, gestione aziendale, contabilità, fatturazione, magazzeno, paghe e contributi, wordprocessing.
Il sistema è completo di interfaccia seriale RS-232 per collegamento a una stampante e inoltre si possono collegare altri 3 registratori a cassetta.
Entro fine anno saranno anche disponibili i floppy-disk con la possibilità di collegare fino a un massimo di 4 driver.
Il Mistral 801 é distribuito in Italia dalla P.B.S. di Milano.

P.B.S. MILANO

TERMINALI

Videoterminali a microprocessore

La serie di videoterminali Infoton comprende tre modelli, di cui due basati sul microprocessore Z-80.
Nella fascia inferiore si colloca il modello I-100, un video terminale con schermo da 12" a 25 righe per 80 colonne.
Accanto a funzioni riscontrabili in quasi tutti i dispositivi CRT, l'I-100 consente in più un editing locale molto potente, terminali con tasti per funzioni definibili dall'utente, una porta di collegamento per hardcopy visualizzazione nero su bianco e bianco su nero, trasmissione selezionabile a carattere o a pagina, nonchè possibilità di tracciamento e visualizzazione di grafici.
Di rilievo nel modello Infoton 100 anche la possibilità di emulare gran parte dei videoterminali oggi in commercio.

All'estremo superiore si colloca invece il modello I-400, che, accanto alle caratteristiche proprie dell'I-100, ha in più la tastiera svincolabile dal corpo del visore, due porte per il collegamento di stampanti - di cui una seriale e l'altra parallela - collegamento di più terminali interrogabili via polling, visualizzazione nero su bianco per campi singoli nonchè buffer con capacità di memoria pari a due pagine di schermo.

DATA PERIPHERAL ITALIANA-MILANO

Controllo della produzione nell'industria manifatturiera

La Hewlett-Packard ha presentato una serie di supporti informatici per la gestione in tempo reale delle operazioni di ricevimento merci, dei controlli di inventario, della produzione e della qualità, della gestione dei magazzini e delle spedizioni, del controllo degli orari di lavoro, della fatturazione e della contabilità.
L'HP 3077A è un terminale specializzato per il controllo delle presenze, montabile a parete, con un lettore di badge Type V od un lettore opzionale multifunzione; se necessario, possono essere collegati fino a 56 di questi terminali, con varie configurazioni, ad un singolo sistema HP 1000, e gestiti mediante il DATACAP/1000, attraverso uno più collegamenti, per trasmissioni dati. Ognuno di questi può essere lungo fino a 8 km con l'impiego di un cavo singolo.

Per i terminali più lontani si possono usare collegamenti telefonici.
Il sistema HP controlla automaticamente ogni introduzione nel momento in cui viene effettuata, ed avverte l'utente se vi sono delle omissioni o degli errori evidenti.
Il software DATACAP/1000 mette a disposizione dell'utente del sistema la possibilità di specificare le transazioni di introduzione dei dati necessarie per la particolare organizzazione, mediante un semplice processo interattivo su un terminale video.
Per i reparti di produzione e i magazzini, la Hewlett-Packard ha messo a punto anche i terminali HP 3075A, montato a parete, e HP 3076A, in versione da tavolo. Ognuno di questi guida l'operatore con indicazioni luminose che possono essere configurate dall'utente con una propria terminologia.

HEWLETT-PACKARD

Un nuovo terminale video dalla Informer

È stato presentato dalla Microlemdata all'EDP USA '79 il nuovo Terminale Video della Informer Inc. modello D-304.
Per le sue prestazioni ed il gradevole design ha riscontrato il favore di molti operatori del settore.
Permette tra l'altro, la rotazione di 350° dello schermo per ottenere la miglior posizione o per mostrare ad altri il contenuto dello schermo stesso; permette di avere di fronte una persona e il teminale allo stesso empo senza nient'altro di interposto che la tastiera, oppure dà la possibilità di appoggiare i documenti su di un supporto proprio di fronte all'operatore con conseguente alleggerimento del lavoro di ricerca visiva e maggior resa; può essere richiesto inoltre anche con lo schermo verde.
Tecnicamente parlando, le possibilità del Polled Block Mode (multi terminal su single line), di selezionare da software quattro formati (80×24 - 64×16 - 32×16 - 40×12) mantenendo 2K di memoria in scroll, I/O di dare attributi ad ogni carattere, di avere 28 funzioni, di selezionare da software il block o l'interactive mode, dell'autorespect su

ogni tasto, dell'uscita video composito per un monitor parallelo, di baud rates da 50 a 19.200, di diversi modi di cancellazione, del trap mode, della printer option per hard copy, ecc., fanno del D-304 un grande terminale anche se mantiene piccole dimensioni ed un prezzo contenuto.
La struttura di base del D-304 prevede già l'applicazione di lettore di barre e di penna ottica e di parte semigrafica per controllo di firme.

MICROLEMDATA-Vimercate (MI)

SOFTWARE

Il software "ASYNCH" permette il collegamento fra sistemi Zilog e altri calcolatori

Un nuovo package di software offerto dalla Zilog permette agli utenti di qualsiasi microcalcolatore MCZ della Zilog di comunicare con altri MCZ e con molti altri calcolatori di tipo "mainframe" attraverso un collegamento seriale come quello usato per i terminali.
Tale insieme di programmi, denominato "ASYNCH", viene chiamato attraverso il sistema operativo "RIO", ed è ideale per piccoli utenti gestionali che richiedano una capacità di elaborazione distribuita. Per esempio, gli uffici distaccati di una società possono elaborare indipendentemente le informazioni sulla situazione locale del magazzino, ecc., e poi trasmetterle ad un MCZ centrale alla fine del periodo di lavoro prestabilito.
L'ASYNCH permette anche di ridurre i tempi di collegamento ed i costi d'uso di altri calcolatori, consentendo ai programmatori di scrivere, compilare e mettere a punto i loro programmi usando come posto di lavoro un MCZ, e trasferendo poi il testo del programma finito ad un calcolatore principale.
Si ricordi a questo proposito che sono a disposizione per tutti gli MCZ i compilatori COBOL ANSI (livello 1 completo e parte del 2), e FORTRAN IV, il che li rende del tutto adatti come strumenti di programmazione a basso costo per questi due linguaggi.
Il package ASYNCH consiste di due parti: un driver (RIO-compatibile) per l'hardware ($SIBDRIVER), ed un emulatore di terminale per gestire protocolli asincroni (COMM).
Il $SIBDRIVER è un driver che permette all'utente di scrivere programmi che controllano il colloquio con altri sistemi. Esso opera su una porta seriale asincrona, che negli MCZ è rappresentata da una scheda (SIB) di interfaccia seriale, e nei sistemi di sviluppo da una scheda (ASPIO) di I/0 seriale e parallelo.
L'input dalla porta seriale è pilotato da interrupt, e i caratteri sono memorizzati in un buffer circolare interno di 256 caratteri.
Può essere configurato per mezzo di semplici chiamate al sistema operativo.
Usando il $SIBDRIVER come interfaccia, il COMM permette ad un operatore seduto alla console di un sistema Zilog di accedere ad un calcolatore "host", o di trasferire dei files di dati e di programmi simbolici da dischetto al sistema principale; è inoltre in grado di effettuare trasferimenti di dati dal sistema "host" a stampante locale.
Una caratteristica speciale del COMM è di poter definire i parametri che interessano la trasmissione (per es. frequenza di trasmissione, parità, numero dei bit di dati, ecc.) attraverso comandi ad alto livello. Il COMM permette anche di trasferire files contenenti programmi oggetto fra sistemi Zilog, in modo da individuare gli errori di trasmissione e ritrasmettere i dati sbagliati.

ZELCO-MILANO

MicroNOVA con linguaggio Pascal

La Data General Corporation ha lanciato un sistema operativo in multitasking ed in tempo reale per i microcalcolatori microNOVA.
Contemporaneamente, è stato annunciato, sempre per la linea microNOVA, il linguaggio evoluto di programmazione Pascal.
Il sistema MICRON permette di incrementare la produttività del programmatore per mezzo del linguaggio Pascal e degli strumenti e programmi di utilità che consentono lo sviluppo e la messa a punto interattiva dei programmi applicativi.
Inoltre, il sistema operativo MICRON è compatibile con il Sistema Operativo Avanzato (AOS) della Data General, utilizzato dai più grandi sistemi ECLIPSE.
Per le attività di sviluppo; il sistema MICRON opera sul microNOVA originale e sui sistemi MP/200 con 64K byte di memoria e sottosistema a disco o dischetto. In fase esecutiva, il sistema MICRON opera su qualsiasi microprocessore microNOVA a 8K byte di PROM e/o RAM, sia esso un chip, una scheda, o un sistema completo.
Con il MICRON sono disponibili memorie PROM e EPROM. Questo consente agli utenti di creare programmi permanenti in memoria centrale per applicazioni che non richiedono periferiche magnetiche.
Il MICRON esegue automaticamente controlli di consistenza su istruzioni e dati durante il tempo di riposo.
Ha una struttura interna modulare che consente all'utente di scegliere le sole ruotines necessarie per le sue elaborazioni senza il sovraccarico di tutto il sistema operativo.
Inoltre assicura l'integrità degli archivi su disco per mezzo del suo programma di utilità fixup che verifica automaticamente la struttura del disco nel caricamento iniziale del sistema e permette eventualmente il ripristino.
Per quanto riguarda il linguaggio strutturato MP/Pascal esso gira sotto il sistema operativo MICRON utilizzato dagli elaboratori microNOVA e sotto l'AOS utilizzato dai sistemi ECLIPSE della Data General.
La compatibilità tra l'MP/Pascal sotto MICRON e sotto AOS permette agli utenti di sviluppare programmi in MP/Pascal che girano sul più piccolo elaboratore a singola piastra microNOVA MBC/1 come pure sul più grande sistema ECLIPSE/600.

DATA GENERAL-Cinisello B. (MI)

Micro Processor Pascal Sistema di Programmazione

La Texas Instruments Semiconduttori Italia annuncia un nuovo prodotto software della famiglia Pascal: il Microprocessor Pascal (MPP).
L'MPP è il più avanzato sistema di programmazione con linguaggi ad alto livello per le applicazioni industriali dei microprocessori in grado di soddisfare completamente le esigenze dei progettisti che vogliono un software portatile ed economico facilmente sviluppabile su un sistema a floppy disk.

Il sistema software MPP utilizza un linguaggio che contiene come sottoinsieme il popolare linguaggio Pascal, permettendo all'utilizzatore di sfruttare a pieno e di non rendere obsoleti i suoi investimenti software.
L'MMP sfrutta le tecniche della programmazione strutturata del Pascal ISO Standard per ottenere programmi modulari, facilmente comprensibili, con esecuzione del software residente su memorie ROM.
Il Pascal dell'MPP implementa la tecnica di esecuzione concorrente di programmi scritti per microprocessori in un ambiente multitask.
Il sistema MPP consiste di 6 parti, includendo le più avanzate specifiche dei traduttori Pascal disponibili attualmente sul mercato:

- Editor, di programmi sorgenti Pascal, progettato specificatamente per la programmazione strutturata. L'Editor include un "analizzatore sintattico" che abbrevia il tempo di sviluppo dei programmi.
- Compilatore, traduce in codice intermedio sia i programmi Pascal convenzionali che le estensioni concorrenti della Microprocessor Pascal della Texas Instruments. Il codice intermedio può essere eseguito direttamente o convertito in codice oggetto per i sistemi della famiglia 9900.
- Debugger, consente l'uso di oltre 15 opzioni per verificare, eseguire, tracciare e modificare programmi sviluppati.
- Configuratore, ritaglia i programmi includendo negli stessi solo moduli di libreria necessari per l'esecuzione sul sistema finale.
- Generatore di moduli oggetto, converte il codice intermedio (P-code) in linguaggio macchina della famiglia del 9900.
- Interprete o supporto run-time, consiste in programmi in grado di supportare i due modi di esecuzione di un programma Pascal (P-code, object code); questo conferisce al progettista un'enorme flessibilità nell'ottimizzazione dei parametri velocità di esecuzione ed occupazione di memoria di un programma.

I sistemi necessari per sviluppare un'applicazione con il sistema di programmazione MPP sono:

- requisiti di un sistema minimo
  FS990/4 sistema di sviluppo per microprocessori a floppy disk
  - 56K bytes di memoria
  - 911 o 913 terminale video
- requisiti per un ambiente multiutente
  DS990 minicomputer multi-utente basato du dischi rigidi
  - 64K bytes di memoria

Sistemi che eseguono programmi Pascal con esigenze di risposta in tempo reale sono:
9900 microprocessori a 16 bit
- TM990 moduli microcomputer con estensione di memoria
- 990 minicomputer.

TEXAS INSTRUMENTS SEMICONDUTTORI ITALIA

Compilatore native Pascal

La Digital Equipment ha reso disponibile un compilatore Pascal realizzato per applicazioni industriali e di addestramento utilizzabile su sistemi VAX-11/780.
VAX-11/PASCAL è un compilatore rientrante, scritto in modo "native", che sfrutta i vantaggi offerti dal set di istruzioni in virgola mobile e di manipolazione caratteri conché dalla capacità di memoria virtuale del sistema operativo VAX/VMS.
Il prodotto offre le capacità standard del linguaggio Pascal compresi i tipi di dati strutturati e scalari, il controllo di loop e le istruzioni condizionali, le procedure di I/O e altre funzioni standard.
VAX/11/PASCAL comprende inoltre altre innovazioni interessanti. Fra queste figurano i files sequenziali di records di lunghezza fissa o variabile, un tipo di dato reale a doppia precisione e una parte di dichiarazione valore.
Altre caratteristiche standard sono rappresentate dalla compilazione separata di moduli, un'interfaccia standard di richiamo per programmi scritti in altri linguaggi di programmazione nativi VAX/VMS, e il completo accesso a tutti i servizi del sistema operativo VAX/VMS.
Infine, VAX-11/PASCAL è compatibile anche con le specifiche di normalizzazione ANSI X3J9 attualmente in vigore.

DIGITAL EQUIPMENT-Cinisello B. (MI)

CONCORSO

Aperte le iscrizioni al dodicesimo concorso Philips per i giovani ricercatori europei

Conclusa ad Oslo la undicesima edizione del Concorso Philips per i giovani ricercatori europei, con la premiazione - tra gli altri - di un giovane studioso italiano, gli organizzatori ricordano che sono aperte le iscrizioni per il dodicesimo concorso.
Questa iniziativa è riservata ai giovani di età compresa tra i 12 e i 21 anni, nati fra il primo gennaio 1959 e il 31 dicembre 1968, che si interessano di scienza e di tecnica, ed è patrocinato dal Ministero della Pubblica Istruzione e dal Consiglio Nazionale delle Ricerche.
Al Concorso Philips possono partecipare lavori di ricerca e di innovazione, così come le invenzioni, in tutti i settori dello scibile. La Giuria, composta da eminenti personalità del mondo accademico nazionale, prende infatti in considerazione tutti i lavori presentati, anche di natura teorica ed interdisciplinare, purchè sviluppati ordinatamente e corredati dalle indicazioni dei mezzi adottati e dei risultati ottenuti (misure, controlli, documentazione dimostrativa delle indagini compiute).
A proprio insindacabile giudizio la Giuria potrà assegnare fino a tre primi premi da 500 mila lire ciascuno, cinque secondi premi da 250 mila lire e cinque premi di merito da 150 mila lire. Il termine per l'invio dei lavori dei partecipanti scade alla mezzanotte del 31 dicembre 1979. La cerimonia di premiazione nazionale avrà luogo a Milano nel marzo 1980.
Per questa edizione i giurati, sempre con decisione insindacabile, potranno selezionare (tra coloro che non abbiamo mai preso parte in precedenza alla finale internazionale) un massimo di tre concorrenti ritenuti meritevoli di partecipare, assieme ai finalisti del concorso bandito contemporaneamente nelle principali nazioni europee, alla finale che avrà luogo in Olanda a fine maggio.
Gli interessati possono chiedere regolamento, scheda di iscrizione ed ogni ulteriore informazione alla Segreteria del *Concorso Philips per i Giovani Ricercatori Europei, Piazza IV Novembre, 3, 20124 Milano. Telefono 69.94.359.*

Microprocessor Books

Vol. 0
The Beginner's Book

Questo libro è dedicato ai principianti in assoluto.
Chi ha visto i computer solo alla TV o al cinema può iniziare con questo libro che descrive i componenti di un sistema microcomputer in una forma accessibile a tutti.
Il volume 0 prepara alla lettura del Volume 1.

circa 300 pagine **L. 12.000** (Abb. L. 10.800)

Vol. 1
Basic Concepts

Il libro ha stabilito un record di vendita negli Stati Uniti, guida il lettore dalla logica elementare e dalla semplice aritmetica binaria ai concetti validi per tutti i microcomputer. Vengono trattati tutti gli aspetti relativi ai microcomputer che è necessario conoscere per scegliere o usare un microcomputer.

circa 400 pagine **L. 13.500** (Abb. L. 12.150)

Vol. 2
Some Real Microprocessors

Tratta in dettaglio tutti i maggiori microprocessori a 4-8 e 16 bit, disponibili sul mercato. Vengono analizzate a fondo più di 20 CPU in modo da rendere facile il loro confronto e sono presentate anche le ultime novità, come l'Intel 8086 e il Texas Instruments '9940.
Oltre ai microprocessori sono descritti i relativi dispositivi di supporto.
Il libro è a fogli mobili ed è **fornito con elegante contenitore.** Questo sistema consente un continuo aggiornamento dell'opera.

circa 1400 pagine **L. 35.000** (Abb. L. 31.500)

Vol. 3
Some Real Support Devices

È il complemento del volume 2. Il primo libro che offre una descrizione dettagliata dei dispositivi di supporto per microcomputers.
Fra i dispositivi analizzati figurano:
Memorie, Dispositivi di I/O seriali e paralleli, CPU, Dispositivi di supporto multifunzioni, Sistemi Busses. Anche questo libro è a fogli mobili con **elegante contenitore** per un continuo aggiornamento. Alcune sezioni che si renderanno disponibili sono: Dispositivi per Telecomunicazioni, Interfacce Analogiche, Controllers Periferici, Display e Circuiteria di supporto.

circa 700 pagine **L. 20.000** (Abb. L. 18.000)

8080 Programming for Logic Design
6800 Programming for Logic Design
Z-80 Programming for Logic Design

Questi libri descrivono l'implementazione della logica sequenziale e combinatoriale utilizzando il linguaggio Assembler, con sistemi a microcomputer 8080-6800-Z-80.
I concetti di programmazione tradizionali non sono né utili né importanti per microprocessori utilizzati in applicazioni logiche digitali; l'impiego di istruzioni in linguaggio assembler per simulare package digitali è anch'esso errato.
I libri chiariscono tutto ciò simulando sequenze logiche digitali. Molte soluzioni efficienti vengono dimostrate per illustrare il giusto uso dei microcomputer. I libri descrivono i campi di incontro del programmatore e del progettista di logica e sono adatti ad entrambe le categorie di lettori.

circa 300 pagine cad. **L. 13.500** (Abb.L. 12.150)

8080A/
8085 Assembly Language Programming
6800 Assembly Language Programming

Questi nuovi libri di Lance Leventhal sono "sillabari" nel senso classico della parola, del linguaggio assembler.
Mentre con la serie Programming for Logic Design il linguaggio Assembler è visto come alternativa alla logica digitale, con questi libri il linguaggio Assembler è visto come mezzo di programmazione di un sistema microcomputer. Le trattazioni sono ampiamente corredate di esempi di programmazione semplice.
Un altro libro della serie, dedicato allo Z-80, sarà disponibile a breve termine.

circa 500 pagine cad. **L. 13.500** (Abb. L. 12.150 cad.)

Some Common BASIC Programs

Un libro di software base comprendente i programmi che riguardano i più diversi argomenti: finanziari, matematici, statistici e di interesse generale. Tutti i programmi sono stati testati e sono pubblicati con i listing sorgente.
Vengono inoltre descritte le variazioni che il lettore può apportare ai programmi.

circa 200 pagine **L. 13.500** (Abb. L. 12.150)

OSBORNE & ASSOCIATES, INC.

Distributore esclusivo per l'Italia:

JACKSON ITALIANA EDITRICE srl

CEDOLA DI COMMISSIONE LIBRARIA - Da inviare a Jackson Italiana Editrice s.r.l. - Piazzale Massari, 22 - 20125 Milano

☐ Spedizione contrassegno più spese di spedizione ☐ Pagamento anticipato con spedizione gratuita.

Nome........................	Vol. 0 - The Beginner's Book	L. 12.000	(Abb. L. 10.800)
Cognome........................	Vol. 1 - Basic Concepts	L. 13.500	(Abb. L. 12.150)
....	Vol. 2 - Some Real Microprocessors	L. 35.000	(Abb. L. 32.000)
Via........................	Vol. 3 - Some Real Support Devices	L. 20.000	(Abb. L. 18.000)
....	8080 Programming for Logic Design	L. 13.500	(Abb. L. 12.150)
C.A.P........................	6800 Programming for Logic Design	L.13.500	(Abb. L. 12.150)
Città........................	Z-80 Programming for Logic Design	L. 13.500	(Abb. L. 12.150)
Data........................	8080A/8085 Assembly Language Progr.	L. 13.500	(Abb. L. 12.150)
Firma........................	6800 Assembly Language Programming	L. 13.500	(Abb. L. 12.150)
Codice Fiscale........................	Some Common Basic Program	L. 13.500	(Abb. L. 12.150)

in vendita presso tutte le sedi G.B.C.

☐ Abbonato ☐ Non abbonato **SCONTO 10% PER GLI ABBONATI**

FEEDBACK

ONORATISSIMO ....

Dopo quello che ha scritto di me il Boss nell'editoriale, sarà meglio che mi presenti per intero, in maniera che la pessima opinione che vi farete di me sia fondata su fatti piuttosto che sulle basse insinuazioni. Dunque: laureato in Fisica (a quel tempo pareva che servisse a qualcosa), ex oceanografo, ex tecnico riparatore di minicomputers, ex insegnante, ex system analist, ex un paio di altre cosette minori, mi occupo di elettronica digitale da dodici anni e di computers da undici. Da sei anni ormai faccio il consulente, cioé vengo convocato presso un'industria che ha un problema risolubile per via elettronica e dopo qualche mese lascio lì un aggeggio costruito da me, contenente sangue, sudore, lagrime ed un pezzetto della mia anima, che risolve il problema; mi consolo di aver abbandonato così questo mio figlioletto elettronico sia rimirando un foglietto di carta con su scritti dei numeri ed il nome di una banca, sia pensando che lo rivedrò non appena andrà in tilt, cosa che prima o poi succede.

Tutto questo lo faccio insieme alla mia piacevole compagna, che nella vita ha fatto più o meno le stesse cose che ho fatto io e che un giorno per amor di pace ho sposato ("ma che pace d'Egitto?" chiederete. Me lo chiedo spesso anch'io).

Ma basta parlare di me, parliamo invece di voi e di questa rubrica. Ci avete scritto, seppellendoci di chili di lettere. Molte riguardano argomenti tanto simili che sembrano scritte di comune accordo, una specie di plebiscito; per esempio, una cinquantina riguardano il picocomputer. Mi perdonerete se ne scelgo una a rappresentarle tutte: su BIT deve rimanere abbastanza spazio anche per gli altri argomenti in programma.

In ogni caso vorrei che ricordaste che questa rubrica è vostra, non mia; risponderò a chi mi scrive (è buona norma di cortesia) e darò il mio parere ed il mio contributo, se potrò farlo; ma la parte significativa rimarranno sempre e comunque le vostre lettere. E le vostre proposte. E le vostre idee. Grazie per aver sopportato questa mia sparata fino in fondo.

Ancora una cosa: fra tecnici è invalso l'uso di darsi del tu; ciò è abbastanza comprensibile, dal momento che lo spagnolesco "Lei" è d'obbligo verso il funzionario, verso l'insegnante, verso l'Autorità, verso lo sconosciuto con cui per il momento non si ha nulla in comune, e denota ormai più sospetto che rispetto. Mentre il "tu" fra quasi sconosciuti ormai significa "ti conosco poco, ma ti rispetto per quello che sei, per quello che sai e per quello che abbiamo in comune". Voi ed io almeno una cosa in comune l'abbiamo: La passione per l'elettronica. È abbastanza.

Fabio Fumi

Memorie e microprocessori

Sono un giovane lettore di BIT, e voglio farvi i miei complimenti per la vostra rivista perché è veramente interessante, e la leggo con piacere. Lo scopo della presente è di chiedervi un piccolo grande favore, con il quale sono certo di interpretare i desideri di molti altri giovani lettori; vi sarei grato se sul prossimo numero di BIT dedicaste la rubrica NUCLEO o altra alla descrizione delle memorie integrate, descrivendone il funzionamento, le caratteristiche come capacità, consumo, velocità, spaziando delle memorie più classiche ai nuovi modelli; insomma, dovreste fare una rassegna di RAM, ROM, EPROM, EAROM, memorie CCD, memorie a bolle, eccetera, e spiegare pure la differenza tra memoria statica e dinamica, a livello di funzionamento. Colgo l'occasione per muovervi due benevole critiche; l'articolo sul linguaggio Pascal pubblicato sul numero 3 mi è sembrato un po' coriaceo, almeno a prima vista; sarebbe meglio cercare di spiegare i linguaggi ad alto livello con un linguaggio "a basso livello", adatto anche ai neofiti, che magari mai avranno un microcomputer che parli Pascal, ma che sono ugualmente interessati. Poi, va bene che l'8080 e lo Z80 sono processori di tutto rispetto, ma mi sembrano un po' privilegiati rispetto agli altri microprocessori. Dato che il microelaboratore a cui aspiro impiega il 6502, che non mi sembra poi tanto male, mi piacerebbe che dedicaste un po' di spazio anche a lui, un po' più di quanto avete fatto sul numero 1.

Francesco Esposito, Chieti.

Caro Francesco,

grazie dei complimenti, ma non esagerare perché a furia di camminare a testa alta per l'orgoglio rischieremmo di cadere in un tombino aperto; sai, abbiamo tutti un amor proprio patologicamente ipertrofico.

L'idea di una rassegna sulle memorie mi pare eccellente, anche perché con l'attuale dovizia di prodotti sul mercato (magari anche solo annunciati e non ancora disponibili) si corre il rischio di perdersi; magari lascerei ad una trattazione a parte i CCD e le bolle, memorie un po' sui generis perché intrinsecamente seriali e quindi meno adatte ad uso immediato su un microprocessore (oltre a tutto per il momento costano un occhio e sono un po' difficili da usare). Passo la proposta a MM (no, il capo di 007 si chiama solo M, non è lui).

Quanto poi alla differenza funzionale fra memorie statiche e dinamiche, penso bastino poche parole: la cella di memoria statica è un flip-flop, quindi fintantoché il chip è alimentato mantiene l'informazione che contiene, salvo volergliela cambiare. Nella cella dinamica invece la sede dell'informazione è un condensatore integrato (quindi di capacità minima) che lavora sul gate di un transistor MOS, portandolo a condurre quando nel condensatore è presente una carica. Come in tutti i condensatori di questo mondo, la carica tende a disperdersi, salvo rigenerarla in qualche modo. Ciò si fa semplicemente rileggendo la cella, dal momento che un paio di elementi ausiliari in ogni cella consentono di rigenerare completamente la carica solo se al momento della lettura questa già c'è. Quest'operazione si chiama "refresh" e va eseguita circa ogni 2 ms per poter essere certi che l'informazione non vada perduta. Visto poi che le celle sono disposte a matrice, qualche diabolico progettista ha pensato di organizzare le cose in maniera tale che, se viene letta una cella, venga rigenerata la carica nell'intera riga di cui questa fa parte; quindi l'uso delle memorie dinamiche, anche se un po' meno diretto che per le statiche, non è poi tanto disperatamente complicato.

Una curiosità: le EPROM e le EAROM assomigliano in un certo qual senso alle RAM dinamiche: anche qui la sede dell'informazione è costituita da una carica, salvo che questa, anziché essere libera, è imprigionata in una zona isolante ad elevata rigidità dielettrica, dove rimane inalterata per anni se non la si vuole rimuovere per mezzo di raggi UV o di un campo elettrico sufficientemente energico.

Veniamo ora alle critiche: siamo d'accordo, un articolo per "addetti ai lavori" rischia spesso di risultare coriaceo ad un neofita; credo che le soluzioni possibili siano o di diventare tutti almeno un po' "addetti ai lavori", o di scrivere quasi solo articoli pesantemente divulgativi. Va da sé che la prima soluzione è di gran lunga preferibile, anche in considerazione del fatto che per entrare nel gergo e nell'ordine di idee richiesti non occorre poi molta fatica, e che parte di questa fatica intendiamo farla noi col proporre una serie di mezzi semplici per entrare nel giro. Ma che cosa intendi per "linguaggio a basso livello"? Sai bene che al mondo non c'è nulla di più testone di un calcolatore, e che il linguaggio a basso livello che capisce lui (sì, siamo tutti un po' animisti e ci piace personificare quei picco-

li infami pezzetti di silicio) è di gran lunga più complicato per un essere umano che, mettiamo, un Basic ad alto livello; questo è una specie di inglese molto semplificato, il primo è una serie di numeri che hanno sì un significato, ma molto ben nascosto. Ma forse tu intendevi un linguaggio umano a basso livello, cioé che dica le stesse cose ma in maniera più semplice; va bene, non è sempre facile, ma ci proveremo.

Seconda critica: anch'io detesto l'8080 e lo Z80, non perché siano cattive macchine, ma perché ci si fa troppo chiasso sopra e quindi sono un po' argomenti d'obbligo (per la stessa ragione non ho la televisione, non seguo le partite di calcio e non andrò a vedere Ratataplan). Bisogna d'altra parte riconoscere che l'Intel è uscita prima degli altri coi microprocessors e che la Zilog, coi poveri mezzi messile a disposizione dalla Exxon, ha fatto un gran chiasso pubblicitario (rischiando di scoraggiare chi sia un po' bastian contrario riguardo all'usare l'ottimo Z80); ciò ha imposto questi due micro sul mercato, rendendoli effettivamente preferenziali, e adeguarsi è fin troppo facile. Per fare ammenda, ed anche perché il 6502 è un eccellente microprocessor, se ne parla in questo numero. Scrivimi ancora, critiche come queste ci sono preziose per aggiustare la mira. Grazie.

Ritardi, dimenticanze e computer high-cost

Sono uno studente in elettronica industriale e ho alcuni appunti da farvi. Per prima cosa la periodicità: non vi sembra di chiedere troppo quando fate aspettare ben tre mesi per leggere il prossimo numero? Scherzi a parte, sarebbe ora che la rivista uscisse mensilmente come le riviste di personal computer americane e che avesse un numero di pagine superiore.

Ho riscontrato che a volte il vostro NEWSLETTER si lascia sfuggire notizie anche importanti; non ci sarebbe il modo di renderlo più informato?

Una cosa della quale forse non siete al corrente è il pessimo servizio di distribuzione di BIT; quando su altre riviste leggo che BIT è in edicola, devo attendere altri 10-15 giorni prima che esca effettivamente.

Comunque devo dire che BIT è già una buona rivista piena di articoli interessanti anche se forse non tutti sono di facile lettura. Vi volevo chiedere ancora qualcosa: perché non fate qualche articolo (o almeno non date qualche notizia su NEWSLETTER) riguardante la robotica, che a quanto vedo sfogliando riviste americane sta prendendo piede in USA anche a livello home & personal? Perché computer che in America costano poche centinaia di dollari (o un migliaio di dollari al massimo) arrivano in Italia a prezzi raddoppiati o triplicati? Ci sono molte spese di importazione o è l'importatore che ci vuole guadagnare troppo?

Marco di Marco, Rimini

Caro Marco,

e ti lamenti di ritardi? Mi è caduto l'occhio sulla data della tua lettera, per scoprire che è rimasta in fondo ad un cassetto (non mio!) per circa sei mesi. Che ci posso fare? Dacché la tirata d'orecchi ha assunto un significato politico, mi rifiuto di usarla; l'unica cosa che posso tentare è mettermi ad abbaiare tanto forte con il (o la) colpevole da seminare il terrore; anche se poi si rimetteranno a dire che ho un cattivo carattere.

Per la periodicità non ti preoccupare più: siamo bimensili, fra un po' se il Diavolo non ci mette la coda saremo mensili. Non chiedere che BIT diventi quindicinale, non ti prenderemmo sul serio. Contento?

Newsletter ogni tanto perde qualche colpo, è vero. Stiamo facendo il possibile perché non succeda; e anche l'impossibile, ma ci vuole un po' di tempo in più.

Sì siamo al corrente che la distribuzione è, ehm, un po' carente (io che vivo a Trieste le riviste me le vado a prendere direttamente in Jackson a Milano; poi le mie mi arrivano a casa con comodo). Tieni conto che nel Paese in cui viviamo non arrivano puntuali nemmeno i moduli delle tasse (grazie al Cielo); questo non a titolo di scusa, ma per spiegarti che spostare un pezzo di carta da A a B, se A e B si trovano in Italia, non è né la cosa più semplice, né la più veloce del mondo. Nei limiti delle nostre possibilità ci stiamo sforzando di migliorare.

Sono anch'io un fedelissimo di Asimov (anche se ogni tanto lo trovo insopportabilmente pomposo quando parla di sé, ma nei grandi ciò è perdonabile), però la U.S. Robots & Mechanical Men non è stata ancora fondata, quindi la scienza della robotica è ancora tutta da definire. Ce ne occuperemo, anche per tentare di definirla.

Perché prodotti low cost negli USA diventano high cost qui? Per diverse ragioni la prima è che siamo un Paese del terzo mondo (parlando in inglese ho sempre qualche incertezza fra "under developement" e "underdeveloped") almeno da un punto di vista tecnologico, ma temo anche da diversi altri. Importare costa scartoffie, tempo, denaro e fatica, lunghissime attese doganali, scioperi eccetera. Quindi l'importazione è scoraggiata di brutto. Quando ciononostante conviene abbastanza, i prodotti vengono importati da diverse Compagnie e si instaura un regime di concorrenza con prezzi accessibili. Quando conviene poco, o solo dal punto di vista dell'immagine, non da quello dell'economia, è abbastanza umano e naturale che l'importatore tenti di cavarne tutto quello che può (finché c'è chi compra), visto che lavora in un regime praticamente di monopolio. Penso che non sia affatto naturale che in certi campi non esista la concorrenza.

Didattica semplice e completa

Sono un vostro lettore, attento ad ogni fase di crescita all'interno del coloratissimo mondo dell'informatica.

La vostra rivista che senza dubbio ha un segmento di clienti ben confezionato ma si prospetta nel futuro il pericolo di una ghettizzazione solo per super-specialisti. Anche se è sempre troppo agevole fare delle critiche io direi che almeno il 25% della rivista dovrebbe possedere un continuum di riferimento soprattutto per quella fetta di utenti ancora potenziali.

Un primo intervento potrebbe essere quello di rendere comprensibile il codice scelto dall'informatica (cioé l'inglese tecnologico). Quindi, ogni numero della rivista potrebbe avere una o più lettere alfabetiche (inglese-italiano) tradotte, commentate dei vocaboli più usati in tutti i momenti informatici (Es: stringhe, files, plotter ecc.). Allora, un primo strumento in mano all'utente sarebbe un vocabolario molto fine che prenderebbe corpo seguendo man mano lo snodarsi della rivista.

Lievi variazioni di stile possono essere date ai mostruosi listings di programma che ogni tanto ci sorbettiamo (oltretutto sono stampati con caratteri non facilmente comprensibili).

Ultimo settore di modifica di quel 25% da rinnovare è la formatica centrata sull'utente (mi scusi il Dott. Cavalcoli) cioé considerare una certa quantità di persone che partendo da zero verranno portate a conoscere, manovrare ed anche acquistare un computer completo (configurazione intorno ai 2.000.000). A questo proposito suggerisco un corso su personal computer; il corso non dovrebbe comunque essere lungo e tedioso (al massimo 1 anno e mezzo); il microprocessore deve essere scelto fra i più completi, sofisticati e documentati come il 6800 o il beneamato 6502.

È chiaro che quanto ho adombrato richiede per voi un grosso sforzo tecnicoeconimico ma, a questo punto, io ritengo che voi sarete gli unici a poterlo fare in maniera altamente scientifica ed attendibile.

Valerio Cappelli, Terni

Caro Valerio,

sì, il rischio dell'isolamento dei superspecialisti c'è ed è reale; ma non parlerei però di ghetto quanto di "turris eburnea", la famosa torre d'avorio in cui si ritirano Coloro Che Sanno per evitare impuri contatti con i miserabili mortali. Proprio per questa ragione, e ricordando che dalle torri d'avorio si può cadere facendosi anche male, è nato BIT. D'altra parte un linguaggio tecnico, stringato e preciso, è necessario in tutti i campi: prova un po' a pensare cosa succederebbe se, in luogo di "appendicectomia", i chirurghi cominciassero a parlare di "taglio di quell'affare a forma di grosso verme che pende nella parte bassa della pancia"; temo che su pazienti maschi si potrebbe verificare penosi equivoci.

L'unica via d'uscita che vedo è quella che suggerisci, cioé di mettere tutti i lettori di BIT in grado di parlare correntemente il gergo tecnico necessario a scambiarsi informazioni digitali con la necessaria precisione e concisione. Non che ci sia una qualche possibilità di scelta sul linguaggio da usare: l'informatica in senso lato si è sviluppata negli Stati Uniti ed i termini provengono quindi dall'inglese. Questa minestra o quella finestra.

BIT uscirà quindi con un dizionarietto Italiano-Computnik e viceversa, a puntate per non ingolfare la rivista.

Sì anche alla proposta di sviluppare un corso per computer, ma diamo tempo al tempo, per il momento non siamo ancora pronti: lo sforzo, come dici, è notevole e si tratta di un lavoro che ha senso solo se fatto bene. Opinioni e suggerimenti in merito sono bene accetti.

Nota: *Rispondo ad una piccola parte delle lettere che sono arrivate, debordano comunque ampiamente dallo spazio che i Grandi Capi di BIT intendevano concedermi. Voialtri che scorrete Feedback cercando inutilmente il vostro nome, perdonatemi e trovate la pazienza di aspettare un po'. In ogni caso i vostri suggerimenti, anche se non c'è lo spazio di parlarne qui, li abbiamo letti e li stiamo digerendo pian piano (ma che cosa pretendete, non siamo Mandrake!).*

Precisazioni

Con la presente Vi invitiamo quanto prima ad eseguire opportuna rettifica dell'articolo apparso sulla Vostra rivista di settembre/ottobre 1979 n° 4 pagina 23.

In esso viene menzionata la Ditta HOMIC quale distributrice esclusiva in campo nazionale del Micro-Computer TRS-80 e questo è un grave errore poiché la Homic è solo nostra Concessionaria a carattere regionale.

L'innesatta dichiarazione, per quanto banale potrebbe sembrare, danneggia altri esistenti Concessionari di zona, creando tra l'altro dannosi equivoci in quella che è la nostra continua ricerca di collaboratori e negozianti.

Vogliate pertanto provvedere alle necessarie ed opportune modifiche per l'interesse di entrambi, diffidandovi nel contempo dall'incorrere nuovamente comunque e dovunque in analogo errore.

Onde evitare che ci si rivolga ad Aziende non autorizzate, pensiamo sia opportuno pubblicare quanto prima, come unico distributore nazionale:

RADIO SHACK ITALIA
20100 MILANO - Corso Europa 12, tel. 02/656093 - 702406
70100 BARI - Via Melo 199, tel. 080/237786

Distinti saluti.

Ing. M. G. MANNARELLI
(Italian Director)

La lettera dell'Ing. Mannarelli merita alcune precisazioni:

1) *L'Ing. Mannarelli non ha (stranamente) rilevato che l'articolo in cui è riportata l'affermazione a cui è fatto riferimento, è in realtà il testo di un'intervista con l'Ing. Franchini della Homic, come peraltro è citato in fondo all'intervista stessa. Quindi la "diffida", come ama esprimersi l'Ing. Mannarelli, sarebbe stato esatto inviarla alla Homic, e per conoscenza anche a noi, cosa che dal testo della lettera non compare assolutamente.*
2) *È costume che le iniziative commerciali, onde evitare malintesi di qualunque genere, siano pubblicizzate sugli opportuni organi di stampa, in forma di comunicato stampa o analoga. Ora non ci risulta cha la Radio Shack Italia dell'Ing. Mannarelli abbia usato tale forma nei confronti di chichessia; e noi non pensiamo d'altra parte che la sua lettera rientri nella categoria suddetta.*
3) *La Homic ha fatto le dichiarazioni di cui sopra in occasione della mostra - convegno BIT 79, cioé in data 6-9 giugno. In quell'occasione la Radio Shack Italia non era presente come espositrice, benché BIT 79 fosse la prima rassegna di personal computers. Perché?*
Comunque in seguito a Milano ci sono stati l'EDP USA e, più recentemente, lo SMAU. Anche in queste occasioni la Radio Shack Italia non si è vista.
Meraviglia perciò che siano più attivi i distributori regionali tipó Homic che non il distributore nazionale.
Se da un lato l'Ing. Mannarelli si preoccupa giustamente che non vi sono "equivoci in quella che è la loro continua ricerca di collaboratori e negozianti", BIT, come rivista di appassionati di personal computers, si preoccupa che dietro un nome vi sia effettivamente un'organizzazione al servizio degli utenti.

Marcello Montedoro

Dalla WAVETEK®

Una gamma completa di:

GENERATORI DI FUNZIONE
GENERATORI DI SEGNALE
GENERATORI DI IMPULSI
GENERATORI DI SWEEP
GENERATORI PROGRAMMABILI

con l'assistenza tecnica e il servizio

Via Timavo, 66 - 20099 Sesto S. Giovanni (Mi)
Tel. (02) 2485233 - 2476693 - Telex: 320346

Via G. Armellini, 39 - 00143 Roma
Tel. (06) 5915551 - Telex: 680356

☐ Ricevere un'offerta
☐ Essere inseriti nel Vs. mailing list.
☐ Gradirei la visita di un Vs. Tecnico

NOME .... COGNOME ....

VIA .... CAP. .... CITTÀ ....

TEL. .... DITTA ....

MANSIONI ....